



**FACULTAD DE INGENIERIA U.N.A.M.  
DIVISION DE EDUCACION CONTINUA**

**CURSOS INSTITUCIONALES**

**▪ INTERNET ▪**

del 2 al 6 de octubre de 1995.

**COMISION NACIONAL DE ELECTRICIDAD**

**MATERIAL DIDACTICO**

**Ing. Juan F. Magaña  
México, D.F.**

## TEMARIO INTERNET

### 1.- INTRODUCCION

#### ☐ HISTORIA

#### ☐ DEFINICIONES

#### ☞ RED

#### ☞ HOSTS Y TERMINALES

#### ☞ SISTEMAS CLIENTE/SERVIDOR

#### ☞ X-WINDOW Y CLIENTES X

#### ☐ PRINCIPALES SERVICIOS PROPORCIONADOS POR INTERNET

#### ☞ Servicios

↳ CORREO ELECTRONICO (e-mail)

↳ CONEXION REMOTA (telnet)

↳ NOTICIAS (usenet)

↳ TRANSFERENCIA DE ARCHIVOS (ftp anónimo)

↳ CONVERSACIONES (talk & ytalk)

↳ CONFERENCIAS (internet relay chat 'IRC')

↳ MENUES DE ACCESO (gopher)

↳ REVISTAS ELECTRONICAS

↳ LISTAS DE CORREO

↳ INTERNET BBS

#### ☞ Herramientas

↳ BUSQUEDA DE PERSONAS (finger)

↳ BUSQUEDA DE ARCHIVOS (archie)

↳ BUSQUEDA DE INFORMACION (WAIS)

↳ BUSQUEDA DE DIRECCIONES (white pages)

↳ BUSQUEDA DE RECURSOS (veronica & jughead )

#### Otros

↳ ACCESO INTEGRAL A INTERNET (WWW)

↳ REALIDAD VIRTUAL (MUD)

#### ☐ DIRECCIONES INTERNET

#### ☞ REDES DE CONMUTACION DE PAQUETES

#### ☞ EL PROTOCOLO INTERNET, IP

#### ☞ EL PROTOCOLO DE CONTROL DE TRANSMISION (TCP)

#### ☞ SISTEMA DE NOMENCLATURA DE DOMINIOS

#### ☞ ESTRUCTURA DEL SISTEMA DE DOMINIOS



## CONEXIONES INTERNET

- CONEXIONES TELEFONICAS
- CONEXION DE UNA TERMINAL A UNA LINEA TELEFONICA
- CONEXION DE UNA COMPUTADORA A UNA LINEA TELEFONICA

## **2.- EL CORREO ELECTRONICO Y SUS HERRAMIENTAS**

### CORREO ELECTRONICO

#### INTRODUCCION

#### FUNCIONAMIENTO

- DOMICILIOS
- AGENTES DE TRANSPORTE
- AGENTE DE USUARIO

#### AGENTES DE USUARIO MAS COMUNES

- AGENTES DE USUARIO PARA SISTEMAS UNIX
- AGENTES PARA PC 'S

#### ENVIO DE DATOS BINARIOS POR CORREO

#### FINGER

## **3.- GRUPOS DE NOTICIAS**

### USENET

- INTRODUCCION A USENET
- JERARQUIAS
- LISTAS DE PREGUNTAS MAS FRECUENTES (FAQ)
- INTERPRETACION DE LOS ARTICULOS DE USENET
- LA CABECERA
- LA SIGNATURA
- EL LECTOR DE NOTICIAS COMO UN CLIENTE



## CONVERSACIONES DENTRO DE INTERNET

- EL PROGRAMA TALK
- EL PROGRAMA YTALK

## INTERNET RELAY CHAT (*IRC*)

- FUNCIONAMIENTO

### **4.- TELNET Y *ftp***

#### TELNET: SESION REMOTA INTERACTIVA

- INTRODUCCION
- EL PROGRAMA *telnet*
- PUERTOS VIRTUALES
- COMANDOS DE *telnet*

#### *ftp*: TRANSFERENCIA DE ARCHIVOS

- INTRODUCCION
- EL PROGRAMA *ftp*
- COMANDOS DE *ftp*
- EL PROGRAMA FTPMAIL

#### ARCHIE: LOCALIZACION DE ARTICULOS EN *ftp* ANONIMO

- INTRODUCCION
- UTILIZACION DE ARCHIE CON *telnet*
- UTILIZACION DE LAS BASES DE DATOS WHATIS
- CLIENTES ARCHIE
- ARCHIE POR CORREO ELECTRONICO

### **5.- *gopher*, *veronica* y *jughead***

- INTRODUCCION
- CLIENTES GOPHER •
- ORDENES BASICAS DE *gopher*
- TIPOS DE RECURSOS DE *gopher*
- UNIVERSITY OF MINESOTA DIRECTORY <CSO>
- veronica*
- jughead*



## 6.- World Wide Web

- ✓ INTRODUCCION
- ✓ BROWSERS O VISUALIZADORES

## 7.- OTROS SERVICIOS DE INTERNET

### ☐ TIPOS DE ARCHIVOS

- ✓ INTRODUCCION
- ✓ TIPOS DE ARCHIVOS UTILIZADOS CON *ftp anónimo*
- ✓ TIPOS DE ARCHIVOS UTILIZADOS EN *Usenet* Y EN CORREO ELECTRONICO
- ✓ "BAJAR" ARCHIVOS A NUESTRA COMPUTADORA DESDE INTERNET

### ☐ WAIS

- ✓ INTRODUCCION
- ✓ CLIENTES WAIS
- ✓ EJECUCION DE WAIS
- ✓ TRUCOS PARA ELEGIR UNA FUENTE

### ☐ LISTAS DE CORREO

- ✓ INTRODUCCION
- ✓ TIPOS DE LISTAS DE CORREO

### ☐ PAGINAS BLANCAS

- ✓ INTRODUCCION
- ✓ GOPHER
- ✓ SERVIDORES WHOIS
- ✓ KNOWBOT
- ✓ FRED Y DIRECTORIOS X.500
- ✓ SERVIDOR DE DIRECCIONES USENET
- NETFIND



A finales de los 70, el ARPANET se estaba acercando a su máximo soporte con 256 máquinas conectadas. El protocolo NCP no podía soportar el tráfico de la red, y era tangible que se necesitaba un sucesor de este protocolo.

En 1982, se logró concebir e implementar el diseño final de un nuevo protocolo, denominado IP (Internet Protocol). IP fue uno de los cuatro protocolos que fueron desarrollados, pero todos trabajaban sobre éste. Los otros protocolos fueron User Datagram Protocol (UDP), Transmission Control Protocol (TCP) e Internet Control Message Protocol (ICMP).

En 1985, la National Science Foundation (NSF) propuso extender la red para dar acceso fácil a investigadores y científicos, situando 5 supercomputadoras más a lo largo de Estados Unidos.

Una vez puesto en marcha este proyecto, la cantidad de conexiones excedía la capacidad de la red, aún con el nuevo protocolo IP. En 1987, NSF anunció una sociedad estratégica para solucionar los problemas de la red, anunciando a MERIT Inc., una compañía operadora de redes, a IBM, un gigante de la manufactura de las computadoras y a MCI, la compañía estadounidense de telefonía. Esta sociedad dio sus frutos en el verano de 1988, con la instalación de NSFNET.

Finalmente, en 1990 el ARPANET original se unió a NSFNET, además de otras pequeñas redes que se habían formado, dando como resultado de la unión de este laberinto de redes ahora conocido INTERNET.

No obstante el considerarla INTERNET como una RED de computadoras o un grupo de redes de computadoras interconectadas entre sí es sólo tomar en cuenta el aspecto físico, porque INTERNET en sí, es el mundo de información al que tenemos acceso alrededor del mundo.

Para la mayoría de las personas lo más importante de INTERNET son las personas que en ella participan, que a través de ella se conocen, trabajan y se comunican, para algunas es tan importante o más que el mismo teléfono o el correo postal. INTERNET es el primer foro general y la primera biblioteca general, cualquiera puede participar a cualquier hora, no importa que religión se practique o que tipo de ropas utilicemos, siempre podremos participar en INTERNET.





Quizá lo más sorprendente de **INTERNET** es que no tiene líderes, nadie la gobierna, no existe una única organización que pague el costo. Más adelante comentaremos la forma en que la **INTERNET** está organizada y de algunos de los organismos que se dan a la tarea de tratar de organizar un poco las cosas, pero de momento no hay quién se encargue directamente de su estructura.

## **DEFINICIONES**

Para comprender mejor algunos de los fenómenos que se suceden dentro de **INTERNET**, comenzaremos con establecer algunos conceptos comunes que se manejan dentro de **INTERNET**.

### **RED**

El término **RED** significa dos o más computadoras conectadas entre sí. Hay un gran número de razones para unir las computadoras en redes, pero las dos importantes desde el punto de vista **INTERNET** son:

-  Permitir la comunicación entre las personas
-  Compartir recursos

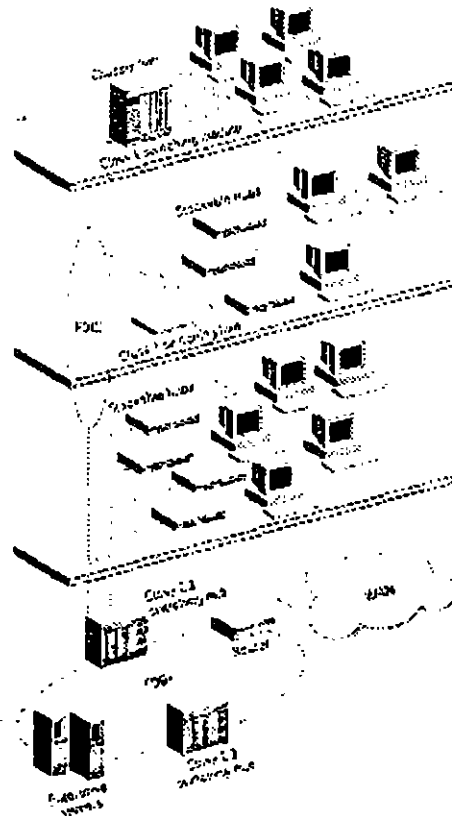
Una vez que estamos dentro de **INTERNET**, podemos enviar mensajes a otras personas que usan otras redes conectadas a **INTERNET**.

En cuanto a compartir, los administradores de sistemas conectan a las redes esos recursos que son caros o difíciles de mantener, haciéndolos disponibles para cualquier usuario de la red. Por ejemplo, un administrador puede conectar a la red una impresora muy costosa de forma que cualquiera que necesite imprimir pueda hacer uso de ella. En **INTERNET** se comparten recursos de información más que dispositivos hardware, esto significa que el tipo de recursos que podemos compartir a través de **INTERNET** son recursos lógicos. Por ejemplo en un catálogo de recursos podemos observar algunos temas que están desarrollados dentro de **INTERNET** alrededor del mundo.

Una red de área local (**LAN**), es una red en la que, las computadoras se conectan directamente a través de un medio de comunicación físico, normalmente con algún tipo de cable. Cuando conectamos unas **LAN** con otras, formamos lo que llamamos una red de área extendida o **WAN**. La mayoría de las redes de área extendida se interconectan utilizando líneas telefónicas que pueden ser dedicadas o conmutadas, aunque hay muchas otras tecnologías, como los enlaces por satélite, que también se usan.



La mayoría de las conexiones de área extendida de **INTERNET** funcionan sobre algún sistema telefónico. En realidad, los cuellos de botella que se producen a veces en **INTERNET** son debidos normalmente a la falta de fiabilidad del sistema telefónico.

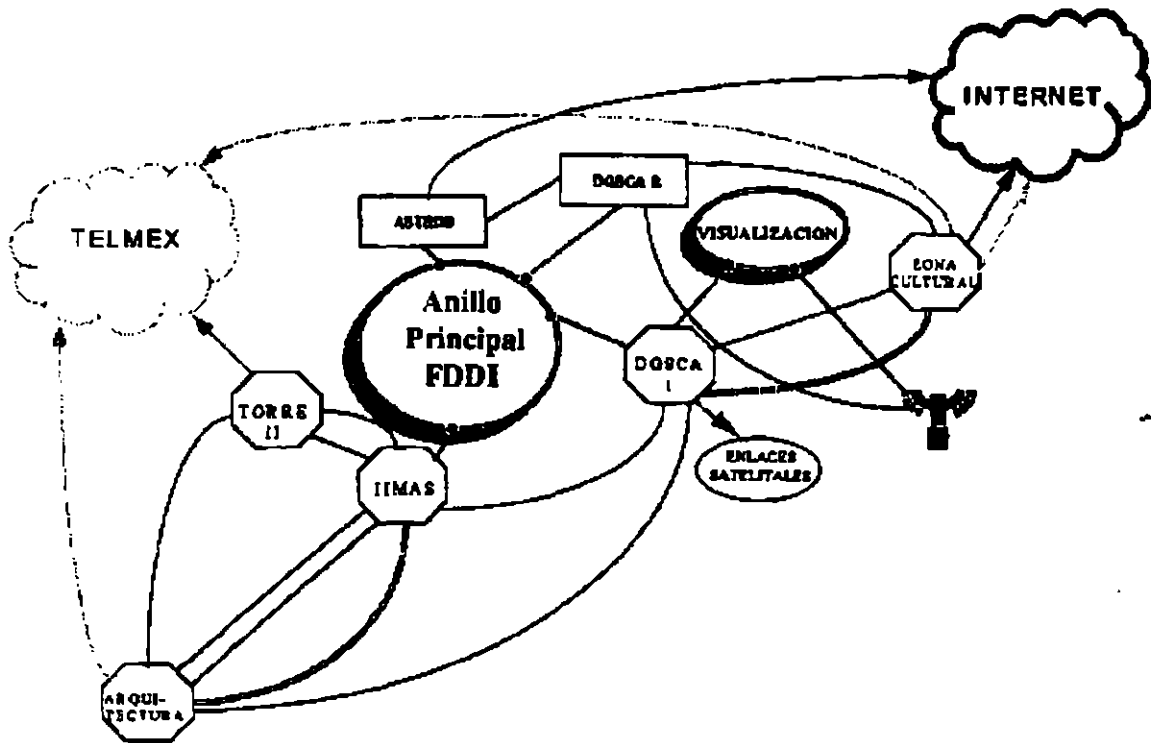


Veamos un ejemplo típico de red. Imagine que está sentado en una sala repleta de computadoras de la facultad de Ciencias Sociales de una universidad, su computadora está conectada a todas las computadoras de la sala y a todas las computadoras del edificio. Ahora bien hay otras LAN en el campus. Por ejemplo, la facultad de Psicología tiene su propia red de computadoras, lo mismo la facultad de Matemáticas, la facultad de Informática y el resto. Cada una de estas LAN están interconectadas entre si a través de un enlace de alta velocidad, para formar una WAN dentro del campus. Aunque hemos tomado como ejemplo una universidad, muchas otras organizaciones utilizan una estructura similar: empresas, gobiernos, centros de investigación, etc. Si la organización es pequeña, puede tener sólo una LAN. Las grandes organizaciones pueden constar de varias LAN, unidas entre si formando una o varias WAN. Estas organizaciones disponen de personal técnico para cuidar del buen funcionamiento de la red.





Para interconectar las LAN se utilizan dispositivos denominados ruteadores, cuya función principal es proporcionar un enlace de una red a otra. Utilizamos ruteadores para unir LAN's y formar WAN, o para unir WAN's para formar WAN's mayores. En otras palabras, se puede decir que las computadoras en **INTERNET** se conectan en LAN's y WAN's mediante un gran número de ruteadores.



Ahora bien, la mayoría de las redes que conforman a **INTERNET** están basadas en plataformas UNIX y algunos de estos sistemas son propiamente multiusuario. No obstante, existen algunos otros conceptos que debemos revisar.

## HOSTS Y TERMINALES

Dentro de **INTERNET** cada computadora por sí sola se denomina host, independientemente de que pueda o no compartir recursos, es decir que la computadora desde la cual accedamos a **INTERNET** es un host, independientemente de que pueda o no poner a disposición de **INTERNET** algún recurso.



Otra de las acepciones que reciben las computadoras dentro de la RED es la de nodo, ya que si se dibuja un diagrama de puntos y líneas para representar las conexiones a una red, cada computadora estará representada por un punto y cada una de las líneas serán las conexiones. La parte de las matemáticas que trata este tipo de diagramas, denomina a cada uno de estos puntos un "nodo". Los especialistas en redes se han apropiado este término para hacer referencia a cualquier computadora conectada a una red. Por eso, "nodo" es un sinónimo más técnico de "host".

La palabra host dentro de los sistemas multiusuario tiene un significado ligeramente diferente, es el nombre que se le da al equipo central del sistema, al cual se le conectan computadoras de capacidades físicas generalmente menores, y que reciben el nombre de terminales.

Una computadora central muy potente puede actuar como host para cientos de usuarios al mismo tiempo. De igual modo una minicomputadora actuará como host para un grupo reducido de usuarios. El sistema operativo UNIX es un sistema multiusuario. Aunque podemos utilizar computadoras Unix como estaciones de trabajo, la mayoría de las computadoras Unix trabajan como host para soportar múltiples usuarios.

En un sistema multiusuario, cada usuario accesa al equipo central a través de su propia terminal, la cual consta de un teclado, una pantalla y quizá de un ratón. Todas las terminales están conectadas al host, que suministra su potencia de cálculo a todo el mundo.

En consecuencia, hay dos acepciones para "host". En **INTERNET**, cada computadora es un host. En un sistema multiusuario, la computadora principal que soporta a cada usuario conectado a través de una terminal, también se llama host. Desde luego, si una de estas computadoras se conecta a **INTERNET**, sería un host multiusuario y un host de **INTERNET**, y cada usuario desde su terminal puede también tener acceso a **INTERNET**.

## ☞ **SISTEMAS CLIENTE/SERVIDOR**

Como sabemos, una de las principales funciones de una red es la posibilidad de compartir recursos. De una forma muy simple, la mayoría de las veces esta compartición de recursos se lleva a cabo por dos programas distintos ejecutándose en computadoras diferentes. Uno de los programas llamado servidor, proporciona un recurso en particular, mientras que el otro programa llamado cliente, utiliza ese recurso y se ejecuta en otra computadora.



En redes de área local, es muy común utilizar la palabra "servidor" para referirse a la propia computadora que ejecuta el programa servidor. En **INTERNET**, normalmente los términos "cliente" y "servidor" hacen referencia a los programas que solicitan y proporcionan los servicios.

Veamos un ejemplo muy común. Muchos nodos de **INTERNET** proporcionan un servicio llamado "*gopher*", el cual permite seleccionar opciones de una serie de menús. Cada vez que se selecciona una opción, *gopher* ejecuta la tarea indicada. Por ejemplo, si la opción del menú describe una determinada información (como "Noticias del Día"), *gopher* recupera esta información y la mostrará en pantalla.

Cuando se utiliza *gopher*, dos programas interactúan entre sí, primero, hay un programa que proporciona una interfaz de usuario, es decir que interpreta las instrucciones que se dan a través de las teclas que se pulsán, muestra los menús y generalmente asegura que las peticiones se lleven al cabo. Este programa se denomina cliente *gopher*. El otro programa es el que suministra cualquier cosa que el cliente *gopher* solicita. Este programa se denomina servidor *gopher*.

Lo importante de este esquema es que los programas cliente y servidor se ejecutan en computadoras diferentes. Por ejemplo, podría estar sentado en su PC en México utilizando *gopher* para leer las "Noticias de Día" de la Agencia de Seguridad Nacional de Virginia, en Estados Unidos. En este caso, el cliente *gopher* se ejecuta en su PC, mientras que el servidor *gopher* es un programa que se ejecuta en una supercomputadora de otro país.

Todos los servicios de **INTERNET** hacen uso de esta relación cliente/servidor. **Aprender a "navegar" por INTERNET, significa aprender a usar cada uno de los programas clientes disponibles.**

**Por esta razón para utilizar un servicio INTERNET, hay que entender**

- ↳ Cómo ejecutar un programa cliente para este servicio
- ↳ Cómo decirle al programa cliente qué servidor se quiere utilizar
- ↳ Qué instrucciones se pueden utilizar con cada tipo de cliente



Lo que el usuario tiene que hacer es ejecutar el programa cliente y decirle lo que tiene que hacer. El trabajo del programa cliente es conectar el servidor adecuado y asegurarse de que sus instrucciones sean enviadas correctamente.

Cada tipo de cliente de **INTERNET** tiene sus propias instrucciones y reglas. Por ejemplo, las instrucciones que pueden utilizar con un cliente *gopher* son diferentes de las que se puede usar con un cliente *archie* (otro servicio de **INTERNET**). La ventaja de los programas clientes de **INTERNET** es que tienen sus propias facilidades de ayuda interactiva.

## ☞ X-WINDOW Y CLIENTES X

Existe un tipo especial de sistema cliente/servidor llamado X-Window, el cual ofrece ciertas ventajas cuando se utiliza **INTERNET**.

El sistema X-Window se utiliza con el sistema Unix y permite soportar interfaces gráficas para el usuario. Una interfaz gráfica de usuario (GUI Grafical User Interface), permite manejar una computadora utilizando además del teclado, un ratón u otro tipo de dispositivo apuntador. Con la ayuda de un ratón, se pueden seleccionar opciones de un menú y manejar objetos en pantalla. Se puede ejecutar más de un programa al mismo tiempo, cada uno de los cuales se ejecutan en su propia área rectangular llamada ventana.

El sistema X-Window fué desarrollado para proporcionar una herramienta estándar para los programadores que desarrollan aplicaciones gráficas y una interfaz estándar a los usuarios para que interactúen con esas aplicaciones.

En la terminología X-Window los tres dispositivos con los que nos comunicamos con la computadora (el teclado, la pantalla y el ratón) se denominan comúnmente display. X-Window nos permite ejecutar más de un programa al mismo tiempo en la misma pantalla. Como parte de la interfaz gráfica de usuario, para cambiarnos de un programa a otro basta utilizar el ratón y desplazarnos de una ventana a otra.

Cuando se utiliza X-Window, la interfaz gráfica de usuario (GUI) para todos los programas que se están ejecutando, son manejados por un programa llamado "display server" o "X server".



Por ejemplo, supongamos que tenemos cuatro programas ejecutándose al mismo tiempo, cada uno dentro de su propia ventana. Mientras trabajamos, podemos mover las ventanas de lugar o cambiar su tamaño. Ahora, supongamos que uno de los programas necesita dibujar un círculo en la pantalla. En lugar de que hacer el trabajo el mismo programa, éste envía un mensaje al servidor X del programa que está controlando la pantalla diciéndole que dibuje un círculo de un determinado tamaño en el lugar especificado. El servidor X ejecuta esa acción.

Esta forma de trabajar tiene muchas ventajas. La primera, significa que todo el entorno gráfico está controlado por un solo programa que garantiza que cada cosa funciona de forma correcta. Por ejemplo, la ventana en la que se está ejecutando un programa puede ser tapada parcialmente por otra ventana. El programa que se ejecuta dentro de la ventana no necesita saber esto, no es su tarea. El servidor X se ocupará de todos estos detalles.

Segundo, cuando un programador diseña una nueva aplicación, no tiene que preocuparse de la interfaz de usuario. Todo lo que se necesita es que el programa llame a las rutinas del servidor X cuando lo necesite. Esto permite hacer programas más pequeños, fiables y que son portables de un sistema X a otro.

Puesto que todos los servidores X proporcionan las mismas funciones, un programa que está escrito utilizando estas funciones, podrá ejecutarse en cualquier sistema X. Por ejemplo, puede encontrar programas X en cualquier lugar de **INTERNET**, copiarlos a su computadora, y ejecutarlos bajo su interfaz gráfica de usuario. Una vez que aprendamos a utilizar los servicios de transferencia de archivos de **INTERNET**, podremos encontrar muchos de estos programas gráficos de forma gratuita.

La parte del sistema X que proporciona el aspecto de interfaz se llama administrador de ventanas. (Técnicamente el administrador de ventanas es un programa que se ejecuta sobre X). Hay dos administradores de ventanas muy utilizados, llamados Motif y Open Look, sin embargo, los conceptos básicos son los mismos sin importar qué administrador de ventanas se utilice. Desde luego, los programas dentro de las ventanas no cambiarán. Básicamente, si se sabe utilizar un administrador de ventanas, se sabrá utilizarlos todos.



Dada la relación cliente/servidor los programas que se ejecutan son clientes y por lo tanto se les denominan Clientes X. Estos clientes X hacen peticiones de servicios al servidor X que se ejecuta en la computadora. En otras palabras, el término "cliente X" es un sinónimo para cualquier programa que se ejecuta bajo un sistema X-Window.

## PRINCIPALES SERVICIOS PROPORCIONADOS POR INTERNET

### CORREO ELECTRONICO (*e-mail*)

Un usuario de **INTERNET**, puede enviar y recibir mensajes de cualquier otro usuario de **INTERNET**. Más aún, puede enviar mensajes a otros sistemas de correo, como pueden ser CompuServe o MCI Mail, que tienen conexiones con **INTERNET**.

Sin embargo, correo electrónico no significa solamente mensajes personales. Cualquier cosa que se pueda almacenar en un archivo de texto puede ser enviado por correo electrónico: programas (fuente) de computadora, anuncios, revistas electrónicas, etc.

Cuando se necesite enviar un archivo binario que no se puede representar como texto habitual, como programas de computadora compilados o imágenes gráficas, existen facilidades para codificar los datos en texto. De igual forma, una vez que se reciben mensajes codificados, es posible decodificarlos para guardarlos con su formato original.

Por eso, desde un punto de vista práctico, se puede enviar por correo electrónico cualquier tipo de archivo a cualquier persona. El sistema de correo electrónico de **INTERNET** es la columna vertebral de la red.

### CONEXION REMOTA (*telnet*)

Se puede hacer *telnet* a cualquier computadora remota de **INTERNET**. Una vez que se ha establecido la conexión, se puede utilizar esa computadora en la forma habitual (si se posee una cuenta válida). Puesto que la mayoría de las computadoras de **INTERNET** utilizan Unix, utilizaremos la misma terminología que Unix.



El nombre de una cuenta de usuario se denomina identificador de usuario (user-id). La palabra secreta que se debe introducir, para comprobar que la cuenta es la suya, se denomina password. Si se posee un identificador de usuario y una palabra clave válidos, se puede conectar con cualquier computadora de **INTERNET**.

Como un servicio público, muchos servicios de **INTERNET** permiten a cualquier usuario conectarse utilizando la cuenta especial *guest*. Por ejemplo, en los Estados Unidos, existe un sistema que proporciona información meteorológica de todo el país. Cualquier persona puede conectar con este sistema y comprobar cuál será el tiempo para el fin de semana.

### **🔍 BUSQUEDA DE PERSONAS (*finger*)**

La mayoría de las computadoras de **INTERNET** tienen una utilidad que permite buscar información sobre un usuario en particular. Este servicio es conocido con el descriptivo nombre de *finger*. En **INTERNET** los usuarios se conocen por su identificador de usuario, se puede utilizar *finger* para encontrar el nombre de un usuario si se conoce su identificador de usuario. Por ejemplo, se puede comprobar que el identificador de usuario "harley" pertenece a Harley Hahn.

Dependiendo de como esté instalado el servicio *finger* en la computadora que utilice, puede encontrar más información sobre la persona que busca, como por ejemplo número de teléfono, dirección de la oficina, y algunas cosas más. Además, algunos servicios *finger* informarán de cuándo fue la última vez que esa persona utilizó la computadora y si tiene correo electrónico sin leer. Esto puede ser muy útil cuando se necesita comprobar si alguien ha recibido un mensaje importante que se le ha enviado.

Hay también formas de configurar parte de la información que otros usuarios ven cuando hacen un *finger* a su identificador de usuario. Se puede incluir la información que se quiera mostrar. Por ejemplo, un profesor puede indicar sus horas de oficina. Alguien que ofrezca una fiesta, puede dar la dirección de su casa. Se puede conocer esta información, sin más que hacer un *finger* a ese identificador de usuario.

También se puede hacer un *finger* a una máquina, en lugar de a un identificador de usuario. En este caso, la computadora responderá mostrando un resumen de todos los usuarios conectados en ese momento.



Por último, algunos sistemas utilizan el servicio *finger* para suministrar alguna información específica. Por ejemplo, en la Universidad de Washington en Seattle hay un usuario al que podrá hacer un *finger* para conocer información de los últimos terremotos.

#### ☞ NOTICIAS (*usenet*)

*usenet* contracción de "User's Network" (red de usuarios) es uno de los principales servicios de **INTERNET**. *usenet* no es una RED. Es un sistema de grupos de discusión en el que artículos individuales se distribuyen por todo el mundo, y actualmente tiene miles de grupos de discusión.

En cada nodo de **INTERNET**, el administrador de la red decide qué grupos de **INTERNET** quiere hacer públicos y qué grupos quiere recibir. Por esta razón, *usenet* no está disponible en todas partes.

#### ☞ TRANSFERENCIA DE ARCHIVOS (*ftp anónimo*)

**FTP** (File Transfer Protocol, Protocolo de Transferencia de Archivos) permite transferir archivos desde una computadora a otra, la mayoría de las veces es desde un host remoto a nuestra computadora, pero también se puede hacer en forma inversa.

El servicio *ftp anónimo* es un servicio público por el cual una organización pone a disposición de todo el mundo una serie de archivos. Podemos acceder a estas computadoras utilizando el identificador de usuario *anonymous*, este identificador de usuario no necesita palabra clave.

El *ftp anónimo* es el servicio más importante de **INTERNET**. Virtualmente cada tipo posible de información está almacenada en algún sitio y está disponible de forma gratuita. Por ejemplo, muchos de los programas utilizados en **INTERNET** son creados y mantenidos por personas o grupos que los distribuyen al mundo entero vía *ftp anónimo*. También se pueden encontrar revistas electrónicas, archivos de grupos de discusión de *usenet*, documentación técnica y muchas cosas más.

#### ☞ BUSQUEDA DE ARCHIVOS (*archie*)

Hay miles de servidores de *ftp anónimo* alrededor del mundo ofreciendo una cantidad inmensa de archivos. El papel de los servidores *archie* es ayudar a localizar donde se encuentra la información que se necesita.





Supongamos que se quiere un determinado archivo, por ejemplo, un programa sobre el que se ha oído hablar. Se puede utilizar un servidor *archie* para que nos indique los servidores de *ftp anónimo* que almacenan ese archivo.

Una vez que se conoce el nombre de estos servidores, se puede utilizar el servicio *ftp* para cargar el archivo.

Si se consideran los servidores de *ftp anónimo* de todo el mundo como una enorme biblioteca mundial, que está cambiando continuamente, se pueden considerar a los servidores *archie* como su catálogo. Realmente, sin los servidores *archie*, la mayoría de los recursos existentes en los servidores *ftp anónimo* serían inaccesibles.

### ☞ **CONVERSACIONES (*talk & ytalk*)**

La utilidad *talk* establece una conexión entre dos computadoras específicas. Una vez establecida la conexión se pueden intercambiar mensajes de forma interactiva, pero la gran virtud de la utilidad *talk* de **INTERNET** es que es posible sostener una conversación con alguien sin importar la distancia que exista entre ellos. La otra persona ve en su pantalla lo que usted escribe, y ambos pueden teclear al mismo tiempo sin que los mensajes se mezclen.

Mientras que la utilidad *ytalk* permite además, la conversación entre más de dos computadoras al mismo tiempo, y lo que se despliega en la pantalla son las líneas que los demás escriben, permitiendo además escribir al mismo tiempo.

### ☞ **INTERNET RELAY CHAT (IRC)**

La utilidad **INTERNET Relay Chat (IRC)** es análoga a la utilidad *talk* pero pueden utilizarla más de dos personas a la vez, se puede tomar parte en conversaciones públicas con un gran número de personas. Estas conversaciones se organizan sobre distintos temas o ideas, se puede utilizar IRC para organizar una conversación privada con las personas que se hayan elegido, de igual forma que una multiconferencia telefónica.

### ☞ **GOPHER**

*gopher* proporciona una serie de menús desde los cuales se puede acceder virtualmente a cualquier tipo de información textual, incluyendo la que proporcionan otros recursos de **INTERNET**. Hay muchos sistemas *gopher* en torno a **INTERNET**, cada uno administrado localmente.



Cada *gopher* contiene cualquier información que las personas que administran el *gopher* local han decidido compartir.

Mientras algunos servidores *gopher* son sistemas aislados, la mayoría de los servidores *gopher* están instalados de forma que pueden conectar con otros servidores *gopher*. Por ejemplo, supongamos que se utiliza un *gopher* en California. Seleccionando una opción de un menú, se puede conectar con otro *gopher* en Africa o en Sudamérica.

Lo que hace al sistema *gopher* tan potente, es que no importa el *gopher* que se esté utilizando ni la información que se utilice, la interfaz de usuario es siempre el mismo sistema de menús.

### ☞ **VERONICA Y JUGHEAD**

Nadie conoce realmente cuántos sistemas *gopher* hay en el mundo. Basta decir que hay muchos, ofreciendo cada uno de ellos su propia serie de opciones de menús que ponen a nuestra disposición información y servicios. -

*veronica* es una herramienta que permite mantener la pista de muchos menús de *gopher* alrededor del mundo. Se puede utilizar *veronica* para realizar una búsqueda y localizar todas las opciones de menú que contienen ciertas palabras clave (cualquiera que se especifique). *jughead* hace lo mismo para un grupo específico de menús de *gopher*.

El resultado de una búsqueda con *veronica* o *jughead* es un menú, el cual contiene todos los elementos resultado de la búsqueda, si seleccionamos algún elemento de este menú, de manera automática nos conectamos con el *gopher* apropiado, dondequiera que esté. De hecho, a menos que se especifique lo contrario, no se sabrá qué computadora se está utilizando ni de que país.

### ☞ **SERVIDORES WAIS**

Los servidores *wais* proporcionan otro método de búsqueda de información que se encuentra dispersa por **INTERNET**. *wais* puede acceder a un gran número de bases de datos, para lo cual es necesario indicarle a *wais* en qué base de datos se quiere hacer la búsqueda, después *wais* buscará cada palabra en cada artículo en todas las bases de datos que se le indiquen.



El resultado de una búsqueda *wais* es una lista de artículos seleccionados de las distintas bases de datos, *wais* presenta entonces un menú desde el cual es posible pedirle a *wais* que muestre cualquiera de los artículos elegidos.

El nombre *wais* proviene de "Wide Area Information Service" (Servicio de Información de Area Extensa).

## ☞ WORD WIDE WEB

El servicio **World Wide Web** a menudo llamado **Web** o **www** es una herramienta basada en hipertexto que permite recuperar y mostrar información basada en búsquedas por palabras clave. Lo que hace al servicio World Wide Web tan potente es la idea de hipertexto: datos que contienen enlaces a otros datos incluyendo imágenes y sonido.

Por ejemplo, cuando se esté leyendo alguna información, aparecerán ciertas palabras y frases marcadas de una forma especial. Se le puede decir a Web que seleccione una de estas palabras, quién siguiendo el enlace, encontrará la información relevante y la mostrará. De esta forma, se puede saltar de un sitio a otro, siguiendo los enlaces lógicos en los datos.

## ☞ DIRECTORIO DE PAGINAS BLANCAS

Dentro del abrumador mundo de **INTERNET**, nada es más importante que la dirección electrónica de una persona. Una vez que se conoce su dirección, es posible enviarle correo, mantener una conversación con *talk* o utilizar *finger* para obtener más información sobre esa persona. Los **Directorios de Páginas Blancas** nos permiten encontrar las direcciones electrónicas de alguien. Sin embargo, no hay un directorio único de **INTERNET**. Más bien, hay una serie de Directorios de Páginas Blancas de propósito especial en los que se puede buscar información sobre personas en **INTERNET**.

## ☞ REVISTAS ELECTRONICAS

En **INTERNET** existe una gran cantidad de revistas que se publican electrónicamente. Esto es, los artículos se almacenan en archivos de texto que son accesibles para todo el mundo, Algunas de estas revistas electrónicas son periódicos sobre investigación de interés principalmente para especialistas en determinadas materias, algunas otras son de interés general.



Hay dos formas de distribuir las revistas electrónicas. Algunas se distribuyen a través de listas de correo. Cuando aparece un número nuevo, éste se envía a todos los suscriptores a través de correo electrónico. Otras revistas se almacenan en servidores de *ftp anónimo* muy conocidos. Pueden cargar los nuevos números en su computadora, incluso los números atrasados, cuando quiera.

## ☞ LISTAS DE CORREO

Una lista de correo es un sistema organizado en el que un grupo de personas reciben y envían mensajes sobre un tema en particular. Estos mensajes pueden ser artículos, comentarios o cualquier cosa relacionada con el tema en cuestión.

Todas las listas de correo -y hay miles de ellas- tienen una persona que se ocupa de mantenerlas. Es posible subscribirse o eliminarse de esa lista, enviando un mensaje a la dirección apropiada. Muchas listas de correo están "moderadas", lo que significa que alguien decide qué mensajes se envían a la lista de correo y cuales no. Otras listas no son moderadas, de forma que se envían todos los mensajes sin censura alguna.

## ☞ INTERNET BBS

Un **BBS**, o tablón de anuncios por computadora (**Bulletin Board System**), es una especie de almacén de mensajes y archivos, a menudo desarrollados para un tema en particular.

Para utilizar un BBS, hay que conectarse y seleccionar opciones de una serie de menús que irán apareciendo.

Típicamente, un BBS es administrado por una persona u organización en particular. Hay innumerables sistemas de BBS en el mundo, la mayoría de los cuales son accesibles por teléfono. En **INTERNET** hay muchos BBS que son accesibles por el más refinado servicio *telnet*.

## ☞ MUD

Un *mud*, o **Multiplic User Dimension**, es una computadora que proporciona una realidad virtual. Para participar en un *mud*, se debe hacer un *telnet* a un servidor *mud*, asumir un papel y explorar, se interactúa con otros usuarios que están representando sus propios papeles.



## DIRECCIONES INTERNET

Los sistemas de redes modernos están contruidos con la filosoffa de "hiveles o capas de servicio", la primera capa mueve bits de un lugar a otro, este nivel se compone de cables y hardware. El siguiente nivel, agrega software básico que permite aislar los problemas del hardware. Para el tercer nivel, se incorpora otra capa de software para dar al software básico algunas características deseadas. Y conforme se agregan capas, agregamos funcionalidad e inteligencia a la red, capa por capa, hasta que obtenemos algo amigable y útil.

## REDES DE CONMUTACION DE PAQUETES

Cuando tratamos de imaginar qué es **INTERNET** y cómo funciona, es normal pensar en un sistema telefónico. Después de todo, ambos son electrónicos, ambos permiten abrir una conexión y transferir información, e **INTERNET** está compuesta principalmente por líneas telefónicas permanentemente dedicadas a este uso. Desafortunadamente, esto crea una idea errónea y provoca mucha confusión sobre la forma en que funciona **INTERNET**. La red telefónica es una red de conmutación de circuitos es decir que cuando hablamos por teléfono, se separa una parte de la red para dedicarla a atender nuestra llamada. Aun cuando no estemos utilizando nuestra parte de la red (por ejemplo cuando la línea está en espera), ésta es inaccesible para otras personas, lo que provoca una subutilización de un recurso muy costoso: la red.

Un mejor modelo para entender el funcionamiento de **INTERNET** mediante analogías, es el servicio de correo, ya que se opera como una red de conmutación de paquetes. Nosotros no contamos con una parte de la red dedicada a nuestras actividades. Lo que queremos enviar se mezcla con los mensajes de otras personas, se pone en un conducto, se transfiere a otra oficina de correos y se clasifica todo nuevamente. Aunque las tecnologías son completamente diferentes, el servicio de correo es sorprendentemente similar, razón por la cual esta analogía la utilizaremos a lo largo del capítulo.

## EL PROTOCOLO INTERNET, IP

Las diferentes partes de **INTERNET** están conectadas por un conjunto de computadoras llamadas *enrutadores o ruteadores*, que interconectan a las redes. La arquitectura de estas redes puede ser Ethernet, Token-Ring, líneas telefónicas o cualquier otro medio físico. No obstante ésta es equivalente a los camiones y aviones del servicio postal. Es el medio a través del cual el correo va de un lugar a otro.



Podemos decir que los ruteadores son las sucursales postales, ya que son los encargados de decidir cómo dirigir la información o "paquetes"; de la misma forma que una oficina de correos decide como distribuir los sobres por correo. No toda subestación o todo ruteador cuenta con una conexión a cada uno de los otros ruteadores de la red. Si envíamos un sobre de correo desde Baja California con destino al Distrito Federal, la oficina de correos no reserva un avión de Baja California a México para llevarlo, sino que envía el sobre a la sucursal de correo más cercana en dirección al destino y ésta a su vez lo envía a otra, y así sucesivamente hasta que nuestro sobre alcanza su destino final. Esto significa que cada subestación sólo necesita conocer las conexiones con las que cuenta y cual es el mejor "siguiente salto" para acercar el paquete a su destino. **INTERNET** trabaja de manera similar, un ruteador se fija en el destino de la información y decide a donde enviarla. El ruteador elige cual es el enlace más apropiado para enviar la información.

Ahora ¿Cómo sabe la red a donde se dirige la información? Si se quiere enviar una carta, no basta con poner el papel escrito en el buzón y esperar a que sea entregado. Es necesario poner el papel con la información en un sobre; escribir la dirección del destinatario y pegar los timbres postales.

De la misma manera que la oficina de correos tiene reglas que definen la operación de su red, también existen reglas que definen la operación de **INTERNET**. Las reglas son llamadas protocolos. El **PROCOLO/INTERNET** (IP) se hace cargo de establecer direcciones o se asegura de que los ruteadores sepan que hacer con la información que les llega. Si comparáramos al protocolo IP con la oficina de correos, éste trabajaría como un sobre, ya que una parte de la información de la dirección va al principio del mensaje. Estos datos le dan a la red información suficiente para hacer la entrega del *paquete*.

Los direcciones de **INTERNET** constan de cuatro números, cada uno menor que 256, y cuando dichos números se escriben, se separan por puntos, como se muestra a continuación:

☎ 125.111.66.5

☎ 101.243.10.6



Las direcciones **INTERNET** están formadas por varios campos, pero dado que **INTERNET** es una RED de redes, los primeros números de la dirección indican a los ruteadores cual es la dirección de la RED a la que pertenecemos, mientras que los últimos números indican cual computadora personal o equipo anfitrión de la red debe recibir el paquete. Bajo este esquema, cada computadora en **INTERNET** tiene un domicilio único.

Si consideramos la dirección "Av. Universidad 1810-A1, Coyoacán, D.F." la parte "Coyoacán, D.F." es como la parte de la dirección correspondiente a la red, la cual le permite al sobre llegar a la oficina de correos correcta, que es la que tiene la información acerca de las calles en un área determinada. La parte "Av. Universidad 1810-A1" es como la dirección del equipo anfitrión; éste identifica a un buzón particular en el área de servicio de la oficina de correos. La oficina de correos concluye su trabajo cuando entrega el correo a la oficina local correcta y ésta lo pone en el buzón correcto. De la misma forma, **INTERNET** concluye su trabajo cuando los ruteadores llevan la información a la red local correcta y ésta entrega dicha información a la computadora personal o equipo anfitrión correctos, localizados en dicha RED.

Por muchas razones prácticas (sobre todo por limitaciones de hardware), la información enviada a través de las redes IP se divide en segmentos de diferente tamaño llamados comúnmente *paquetes*. Y la cantidad de información que puede llevar un paquete normalmente se encuentra entre 1 y aproximadamente 1500 caracteres de largo. Esto previene que cualquier usuario monopolice la red, permitiendo que todos tengan un acceso equitativo. También significa que cuando la red se sobrecarga, su rendimiento afecta a todos los usuarios de la RED en forma distribuida, de tal suerte que desde el punto de vista usuario solamente se torna ligeramente mas lenta.

Una de las propiedades más impresionantes de **INTERNET** es que, en un nivel básico, el protocolo IP es todo lo que necesitamos para participar en la RED. Por supuesto el protocolo IP por si solo no es muy amigable, pero es lo suficientemente útil.

Pese a que con la dirección IP la RED tiene toda la información que necesita para llevar el paquete hasta su destino, Se tienen que resolver varios problemas:

- ↳ La mayoría de las transferencias de información es mayor que 1500 caracteres, estaríamos decepcionados si la oficina de correos sólo entregara tarjetas postales y rechazara cualquier cosa más grande.



- ↳ En ocasiones se presentan errores, ocasionalmente la oficina de correos puede perder una carta. De la misma forma algunas veces las redes pierden paquetes o éstos pueden dañarse durante la transmisión. A diferencia de lo que sucede con la oficina de correos, en **INTERNET** se pueden resolver estos problemas exitosamente.
- ↳ Los paquetes pueden llegar en desorden. Si se envían dos cartas al mismo lugar en días consecutivos no existe la garantía de que viajarán por la misma ruta o llegarán en el mismo orden. Lo mismo sucede con **INTERNET**.

Para corregir todo esto, la siguiente capa de la red nos permitirá enviar grandes cantidades de información y corregirá todas las alteraciones que puedan ser causadas por la red.

### ↳ EL PROTOCOLO DE CONTROL DE TRANSMISION (TCP)

TCP es el protocolo que se menciona frecuentemente junto con el IP y que se utiliza para resolver los problemas mencionados. ¿Qué pasaría si queremos enviar un libro a alguien y la oficina de correos sólo aceptara cartas? Una de las soluciones sería arrancar todas las hojas del libro, ponerlas en un sobre cada una y depositarlas en un buzón. La persona que reciba las cartas tendrá que asegurarse de recibirlas todas y volverlas a empastar en el orden original. Aunque en términos prácticos postales es inviable, esto es precisamente lo que hace el protocolo TCP.

El protocolo TCP toma la información que se desea enviar y la divide en segmentos más o menos a 1500 bytes, enumera cada segmento para que el receptor pueda verificar la información y colocarla en el orden adecuado. Para que el protocolo TCP pueda enviar esta secuencia de números a través de la red, cuenta con su propio "sobre" en el cual escribe la información requerida para su reordenamiento. Un segmento de la información a transmitir se coloca en el sobre del protocolo TCP, el cual a su vez es puesto, dentro del sobre del protocolo IP y una vez que se pone algo en un sobre IP, la red lo puede transmitir.

Del lado del destinatario, una parte del software del TCP reúne los sobres, extrae la información de ellos y la pone en el orden adecuado. Si algún sobre se pierde en la transmisión, el receptor solicita su retransmisión al emisor. Una vez que el protocolo TCP tiene toda la información en el orden adecuado, la pasa a la aplicación del programa que esté utilizando sus servicios.





La descripción anterior del funcionamiento del protocolo TCP es ligeramente utópica. En la realidad, los paquetes no sólo se pierden, sino que además de esto, pueden ser modificados por el mal funcionamiento durante la transmisión a través de las líneas telefónicas o en cualquier otro segmento de la RED, el protocolo TCP también resuelve este tipo de problemas.

Como parte de la información que coloca dentro de su sobre, añade un número de verificación (Checksum) calculado a partir de la información que va a contener cada sobre TCP, y permite que el receptor TCP detecte errores en el paquete transmitido. Cuando un paquete llega a su destino, el receptor calcula el número de verificación y lo compara con el enviado por el transmisor. Si no coinciden, significa que ocurrió un error durante la transmisión. El receptor descarta ese paquete y solicita la retransmisión.

Cuando utilizamos el protocolo TCP, pareciera que tenemos una conexión permanente entre dos aplicaciones, garantizando de esta forma que lo que se transmite de un lado llega al otro. Lo que realmente sucede es que no tenemos un enlace directo entre emisor y receptor, ya que otras personas pueden usar los mismos ruteadores y la misma red de cableado en los lapsos que ocurren entre el envío de cada paquete.

Claro que manejar TCP implica inversión en tiempo, y si la información que requerimos transmitir cabe en un solo paquete (es menor a 1500 Bytes), y no necesitamos garantizar su entrega, podemos utilizar otros protocolos que sean mas sencillos de manejar.

Uno de estos protocolos es *el Protocolo de Datagramas de Usuario o UDP*, el cual substituye a TCP, es decir que en lugar de colocar la información en un sobre TCP para después ponerla en un sobre IP, esta aplicación pone la información en un sobre UDP y después en un sobre IP.

El protocolo UDP es mucho más sencillo, porque no se preocupa porque los paquetes se pierdan, ni por que la información llegue en orden, el UDP se usa comúnmente en programas que envían mensajes cortos y que sólo reenvían la información si no reciben una respuesta en un tiempo determinado. Por ejemplo, queremos desarrollar un programa que busca números telefónicos en una base de datos en algún lugar de la red. No existe ninguna razón para establecer una conexión de TCP con el objeto de transmitir 20 caracteres en ambas direcciones. Sólo es necesario poner un nombre en el paquete UDP, después en un paquete IP y por último enviarlo a la red. La otra aplicación recibirá el paquete, lee el nombre, busca el número telefónico, pone la información en otro paquete UDP y lo envía de regreso.



¿Qué pasa si el paquete se pierde en el trayecto? El programa tendrá que encargarse de esto: si espera por mucho tiempo sin obtener respuesta, simplemente envía otra solicitud.

Ahora que se tiene la habilidad de transferir información entre dos nodos de la red, es posible empezar a trabajar para hacer que **INTERNET** sea más amigable. Esto es posible utilizando el software adecuado para la tarea que se quiera realizar y poniendo nombres con letras en lugar de sólo números para referirse a las computadoras.

## ☞ SISTEMA DE NOMENCLATURA DE DOMINIOS

Al principio, las personas aceptaban que las combinaciones de números como direcciones estaban bien para que las máquinas se comunicaran entre si, pero para los usuarios es preferible utilizar nombres. Por esto, a las computadoras de **INTERNET** se les asignaron nombres para la conveniencia de los usuarios, y las aplicaciones de **INTERNET** permiten el uso de nombres en lugar de una combinación de números para definir las direcciones de las computadoras.

Es un hecho que también los nombres tienen problemas implícitos, debemos asegurarnos de que nunca dos computadoras dentro de **INTERNET** se llamen igual, es necesario tener una forma de convertir los nombres a combinaciones numéricas. Es posible dar un nombre de computadora a un programa, pero es necesario que el programa tenga una forma de convertir el nombre en una combinación de números. Análogamente cuando buscamos en el directorio telefónico, realizamos la misma operación.

En un principio, cuando **INTERNET** era un lugar pequeño, el manejo de los nombres era sencillo. El NIC, Centro de Información de la Red (Network Information Center), estableció un registro, en el cual una persona enviaba una forma electrónica y el NIC incorporaba la información en una lista de nombres y direcciones. Este archivo, llamado *hosts*, era distribuido en forma periódica a todos los nodos de la red. Los nombres eran simples palabras, no se podían repetir. Si se usaba un nombre, la computadora buscaba en la lista y lo convertía a un domicilio numérico. Cuando **INTERNET** creció y se multiplicó, lo mismo sucedió con la tabla de nombres, se requería mucho tiempo para que un nombre quedara registrado y se volvía más difícil encontrar nombres que no hubiesen sido utilizados, también se requería de mucho tiempo de red para poder distribuir el archivo de la lista a cada máquina de la red.



Era obvio que un sistema distribuido en línea era requerido para satisfacer la rapidez con que cambiaba la información del archivo de nombres. A este sistema se le denomina *Sistema de Nomenclatura de Dominios o DNS*.

## ↳ ESTRUCTURA DEL SISTEMA DE DOMINIOS

El Sistema de Nomenclatura de Dominios es un método para administrar nombres distribuyendo en diferentes grupos la responsabilidad de subconjuntos de nombres. A cada nivel de este sistema se le llama *dominio*. Y al igual que a las direcciones IP los dominios se separan por puntos tal y como se muestra en los siguientes ejemplos:

↳ *martini.eecs.umich.edu*

↳ *glis.cr.usgs.gov*

Puede haber cualquier cantidad de dominios en un nombre, al leer un nombre de izquierda a derecha, cada dominio será más vasto que el dominio que tenga a la izquierda. En el nombre *martini.eecs.edu*, *martin* es el nombre del equipo anfitrión, una computadora con una dirección IP. El nombre para esa computadora se asigna y mantiene por el grupo *eecs*, que puede ser el departamento donde ésta se localiza. El departamento *eecs* es parte de la University of Michigan (*umich*). *umich* a su vez, es parte de un grupo nacional de instituciones educativas (*edu*). De esta forma, el dominio *edu* está compuesto por todas las computadoras de las instituciones educativas, el dominio *uiuc.edu* contiene a todas las computadoras de la University of Michigan y así sucesivamente.

Cada grupo puede crear o cambiar todo lo que esté dentro de él. Si *umich* decide crear otro grupo que se llame *ncsa*, lo puede hacer sin solicitar ningún permiso. Sólo tiene que agregar el nuevo nombre a su parte de la base de datos mundial y tarde o temprano todo aquel que lo necesite descubrirá el nuevo nombre *ncsa.umich.edu*, de igual manera, el departamento *eecs* puede comprar una nueva computadora, asignarle un nombre y conectarla a la red sin pedir permiso a nadie. Si cada grupo a partir de *edu* respeta las reglas y se asegura de que los nombres que asigne sean únicos, ningún nombre en **INTERNET** se repetirá.



Hemos dicho que la mejor forma de entender una dirección es leerla de derecha a izquierda. El dominio de primer nivel será la especificación más general. Existen dos tipos de dominios de primer nivel: el formato antiguo *dominios de organizaciones* y el nuevo *dominios geográficos*.

Los dominios de organizaciones están basados en un esquema de direcciones que fué desarrollado antes de que aparecieran las redes internacionales. Fue proyectado principalmente para utilizarse dentro de los Estados Unidos.

La idea era que el dominio de primer nivel debería indicar el tipo de organización que era responsable de la computadora. La tabla 1-1 muestra las distintas categorías. Todas estas categorías existen desde el comienzo de **INTERNET**, excepto *int*, que es de reciente creación para determinadas organizaciones que traspasan las fronteras nacionales (como la OTAN).

Una vez que **INTERNET** se extendió internacionalmente, se hizo necesario crear nuevos dominios de primer nivel que fueran más específicos. Para enfrentarse a esta necesidad, se desarrolló un sistema nuevo de dominios geográficos, en el que una abreviación de dos letras representan un país entero. Hay muchos dominios de primer nivel de este tipo -uno por cada país en **INTERNET**- como ejemplo en la tabla 1-2 aparece un resumen.

**Tabla 1-1 Dominios de primer nivel de tipo organización.**

<b>Dominio</b>	<b>Significado</b>
com	organización comercial
edu	institución educativa
gov	gobierno
int	organización internacional
mil	organización militar
net	gestión de redes
org	organización no lucrativa

Como podemos observar en la Tabla 1-2, los Estados Unidos tienen un dominio geográfico *us*, aunque no se suele utilizar, generalmente ahí se utiliza la nomenclatura organizacional, sin embargo, los dominios geográficos son los que se utilizan en los demás países dentro de **INTERNET**.



Tabla 1-2 Ejemplos de dominios de primer nivel de tipo geográfico.

Dominio	Significado
at	Austria
au	Australia
ca	Canadá
de	Alemania
dk	Dinamarca
es	España
fr	Francia
gr	Grecia
ie	República de Irlanda
jp	Japón
mx	México
nz	Nueva Zelanda
uk	Reino Unido
us	Estados Unidos

Cuando se usa un nombre como *martini.eecs.umich.edu*, la computadora necesita convertir el nombre de la dirección en una dirección numérica, para hacerlo, empieza a hacer peticiones de ayuda a los servidores DNS, empezando por el extremo derecho y recorriendo la dirección hacia la izquierda. Primero, se pregunta por la dirección al servidor DNS local, si el servidor local conoce el domicilio, automáticamente nos conectará hasta la dirección deseada, si no lo conoce entonces el software del servidor local debe comunicarse con el servidor raíz, quien conoce las direcciones de los *servidores de nombres* que tienen a su cargo los dominios de jerarquía superior (los dominios de la derecha, como por ejemplo *edu*). Un servidor de nombres pregunta al servidor raíz la dirección de la computadora responsable de la zona *edu*. Con esta información, se comunica con ese servidor y le pide la dirección del servidor *umich*, después, contacta a esa computadora y le solicita la dirección del servidor *eecs*, finalmente se comunica también con esa máquina y obtiene la dirección de *martini*, que es la máquina con la que originalmente nos deseábamos comunicar.



## ☒ CONEXIONES INTERNET

¿Qué significa tener acceso a **INTERNET**? Significa utilizar una computadora que es parte de una red unida a **INTERNET**. De una forma práctica, significa que se pueden utilizar los recursos de **INTERNET** que se han descrito anteriormente, cuando utilizamos esta computadora, decimos que estamos en **INTERNET**.

Antes de mencionar las distintas formas en las que se puede tener acceso a **INTERNET**, necesitamos conocer las dos formas de conectarse a **INTERNET**. Primero, se puede utilizar una computadora que esté conectada directamente a **INTERNET**, por ejemplo es posible utilizar una estación de trabajo que es parte de una red conectada a **INTERNET**. En este caso la computadora será un *host* de **INTERNET**, con su propia dirección electrónica. La otra forma de conectarse a **INTERNET** es utilizar una terminal (o una estación de trabajo emulando terminal) conectada a un *host* de **INTERNET**. En este caso la propia terminal no está en **INTERNET**, simplemente se hace uso de una terminal que tiene acceso a una computadora que está en **INTERNET**.

Veamos un ejemplo, si dentro de un edificio tenemos un laboratorio de cómputo, en el que hay 40 computadoras conectadas en RED, la cual está conectada a **INTERNET**, entonces todos los usuarios de estas PC's pueden tener acceso directo a **INTERNET**. Ahora, si entramos a la sala de terminales y tenemos ahí 40 terminales, todas ellas conectadas a un equipo central, el cual también está conectado a **INTERNET**, cada usuario de las terminales tendrá acceso a **INTERNET**.

En ambos casos los usuarios tienen acceso a **INTERNET**, la diferencia es que cada PC es un *host* de **INTERNET**, es decir que cada PC tiene su propia dirección **INTERNET** y es autosuficiente, mientras que en la sala de terminales los usuarios acceden a **INTERNET** conectándose a la computadora central que proporciona la conexión a **INTERNET**. De este modo, todos comparten la misma computadora que tiene una sola dirección **INTERNET**.

Lo que ambas soluciones tienen en común es que los dispositivos computadoras y terminales están conectados directamente por algún tipo de cable. Este tipo de conexión se denomina *conexión directa*. La ventaja principal de este tipo de conexión es su permanencia, ya que todo lo que tenemos que hacer es encender la PC o la terminal y la conexión está lista para ser utilizada. Como contraparte la principal desventaja, es la falta de flexibilidad, ya que si queremos cambiar la ubicación física de alguna máquina es necesario desplazar también los cables.



## ☞ CONEXIONES TELEFONICAS

Un sistema mucho más flexible que el esquema anterior es aquel en el que la computadora o terminal utiliza una *conexión telefónica* sobre una línea de teléfono, en estos casos se puede trabajar en cualquier lugar, siempre y cuando se tenga acceso a una línea de teléfono. Para utilizar una conexión telefónica, es necesario un dispositivo de hardware para convertir las señales de computadora en señales telefónicas y viceversa.

En términos técnicos, las señales que envía una computadora son "digitales" y las de una línea telefónica son "analógicas". Un aparato que convierte señales digitales a analógicas se llama *modulador*, mientras que un aparato que convierte señales analógicas a digitales se llama *demodulador*. Cuando se conecta una computadora a través de una línea telefónica, hay que enviar datos en ambas direcciones, razón por la cual utilizamos un *módem* (*modulador -demodulador*).

Existen en el mercado diferentes tipos de modems, sin embargo independientemente del módem que utilicemos para conectarnos, siempre necesitaremos de un módem en cada extremo de la línea telefónica.

## ☞ CONEXION DE UNA TERMINAL A UNA LINEA TELEFONICA

Es muy común, utilizar una PC para acceder a una computadora remota a través de una línea telefónica. Por ejemplo, es posible tener un host de **INTERNET** en el trabajo o en la universidad que se quiera utilizar desde casa. Estos host están normalmente instalados de forma que aceptan conexiones de una terminal. Esto es, se puede conectar una terminal a la línea telefónica (utilizando un módem) y llamar al host remoto. Una vez que se ha establecido la conexión, se puede trabajar en la terminal de la forma habitual.

Dado que la PC es una computadora completa y no meramente una terminal, necesitamos ejecutar en la computadora un programa que *emule* una terminal, ya que el host remoto está configurado de forma que sólo puede comunicarse con terminales. La mayoría de los programas que nos permiten hacer esto son los programas de comunicaciones, los cuales además de inicializar nuestro módem, nos permiten emular diferentes tipos de terminales.

Sin importar el tipo de terminal que emulemos, en lo que debemos tener cuidado es en que sea el mismo tipo que soporta el host, ya que de lo contrario en la pantalla de nuestra máquina veremos solamente caracteres ASCII sin ningún sentido.



Dos de los parámetros a considerar para seleccionar un programa de comunicaciones son el buffer de desplazamiento el cual se refiere a la capacidad de mostrar caracteres que han desplazado hacia arriba en la pantalla, y la posibilidad de trabajarlo al tiempo que ejecutamos otro programa, es decir que lo podamos ejecutar bajo Microsoft Windows o bajo OS/2 por ejemplo, ya que además de poder hacer otras tareas podemos cortar y pegar la información.

## ☞ CONEXION DE UNA COMPUTADORA A UNA LINEA TELEFONICA

La mayoría de las veces, la mejor forma para conectarse a un host de **INTERNET** sobre una línea telefónica consiste en utilizar una computadora que emule una terminal, no obstante debemos recordar que estamos actuando como una terminal, no somos un host de **INTERNET**. En ocasiones tenemos la necesidad de trabajar como un host real de **INTERNET**, en estos casos, existe una solución para establecer una conexión completa a **INTERNET** sobre una línea telefónica.

Primero debemos disponer de algún host de **INTERNET** que actúe como punto de conexión. Entonces instalamos algún programa de comunicaciones que nos permita utilizar el protocolo PPP (Point to Point Protocol) o SLIP (Serial Line **INTERNET** Protocol), los cuales nos permiten enlazar las dos computadoras a través de un módem sin emular terminal, y una vez que se establece la conexión entre las dos computadoras estos programas nos darán capacidades TCP/IP, con lo que nuestra computadora se convertirá en un host real de **INTERNET**, con su propia dirección electrónica oficial.

Cuando se instalan estos sistemas, debemos de tomar en cuenta el tipo de línea telefónica que vamos a utilizar, si usamos una línea conmutada o normal, aunque la computadora sea considerada como un host real de **INTERNET**, no estará conectada todo el tiempo a la RED, por lo que es necesario indicar a la computadora que actúa como punto de conexión, que almacene los mensajes de correo electrónico que llegan cuando no está conectada.

La otra alternativa es utilizar una línea telefónica dedicada o privada, la cual estará permanentemente conectada, con la consecuencia directa de un aumento en el costo, sin embargo, para algunas empresas, una línea telefónica dedicada usando PPP, puede ser una forma relativamente económica para establecer un acceso a **INTERNET**, ya que esta conexión puede proporcionar acceso a **INTERNET** al resto de computadoras de la empresa.





# INTERNET

---



## Cronología

- ↳ 1957 Lanzamiento del satélite Soviético "Sputnik".
- ↳ 1960 El Gobierno de E.U. se preocupa en como mantener la comunicación después de un ataque.
- ↳ 1960 J. LICKLIDER.- Propone interconectar computadoras de gobierno.
- ↳ 1962 Paul Rand (RAND Corporation) sugiere utilizar líneas telefónicas, para transmitir paquetes de Información.
- ↳ 1967 La Universidad de Michigan toma el proyecto ARPANET ( A dvanced Research Projects Agency Network ).

Notas:

# INTERNET



---

## Cronología

- ↳ 1968 BBN ( Bolt Beranek and Newman, Inc), la concesión del proyecto. Dos objetivos: Militar e Investigación.
- ↳ 1969 Universidad de California (Los Angeles) primer Nodo ARPANET.
- ↳ 1969 Universidades de California (Santa Bárbara), Stanford Research Institute y Utah, se integraron como Nodos.
- ↳ 1970 ARPANET tiene 7 Nodos, bajo el protocolo NCP (Network Control Protocol).
- ↳ 1973 25 Nodos, se acumula un mayor número de investigadores y científicos.

Notas:

# INTERNET



---

## Cronología

- ↳ 1979 256 Nodos, el protocolo NCP no soporta el tráfico en la Red.
- ↳ 1982 Se crea un nuevo Protocolo IP (Internet Protocol).
- ↳ 1985 La National Science Foundation (NSF) propone extender la red a más investigadores y científicos, se integran 5 supercomputadoras.
- ↳ 1987 La Red se satura NSF encarga a : Metir, Inc (Compañía Operadora de Redes), e IBM y MCI (Compañía telefónica) el proyecto NFSNET.
- ↳ 1990 ARPANET, NSFNET y Redes pequeñas se unen, para conformar INTERNET.
- ↳ 1994 Comienza Boom Mundial de INTERNET.

Notas:

# INTERNET

---



## Definición

- ↳ Permite la comunicación entre personas
- ↳ Compartir recursos

NOTAS

Notas:

# INTERNET

---



## Definición

En Internet se comparten recursos lógicos

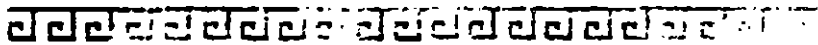
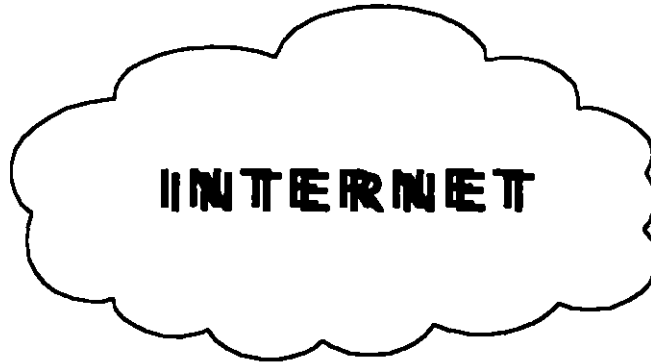
- ↳ Archivos
- ↳ Bancos de Información
- ↳ Noticias
- ↳ Foro de discusiones
- ↳ " Conocimientos "

Notas:

# INTERNET



## Conformación



Notas:

# INTERNET



## Conformación

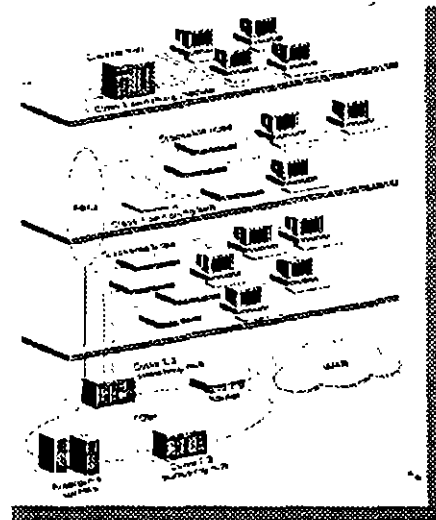
Redes Lan



Redes Wan



Internet



Notas:





# INTERNET

---



## Definiciones

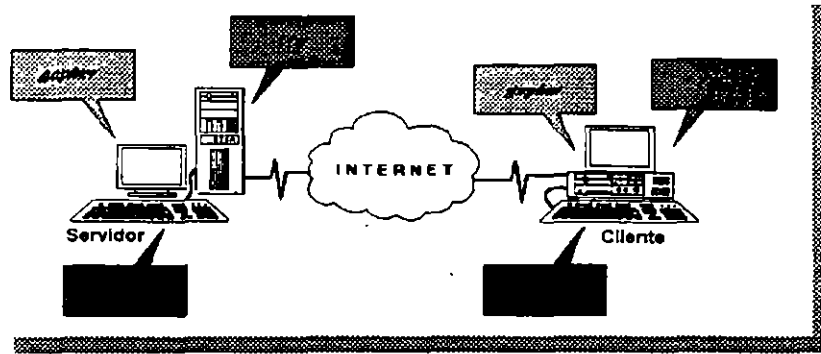
- ↳ **Host**      **Cualquier computadora que accesa a INTERNET.**
- ↳ **Nodo**      **Sinónimo técnico de Host.**
- ↳ **Terminal**    **Teclado y Video conectado a un Host Unix, que a su vez puede ser un Host INTERNET.**
- ↳ **Servidor**    **Host que contiene el software y la información para proporcionar un servicio.**
- ↳ **Cliente**      **Nodo que contiene el software para acceder la información de un servicio de un servidor INTERNET.**

**Notas:**

# INTERNET



## Sistema Cliente/Servidor



Notas:

# INTERNET

---



## Cliente

" Aprender a navegar por INTERNET, significa aprender a usar cada uno de los programas clientes disponibles "

Notas:

# INTERNET

---



## Uso Servicios:

Para utilizar un servicio INTERNET, hay que entender:

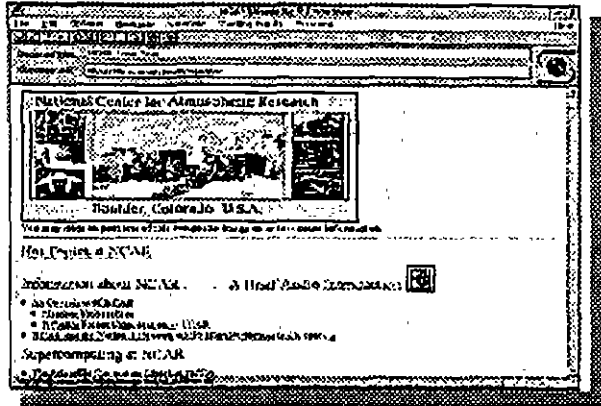
- ↳ Como ejecutar el programa cliente para este servicio.
- ↳ Como decirle al programa cliente que servidor se requiere utilizar.
- ↳ Que instrucciones se pueden utilizar con cada tipo de cliente.

Notas:

# INTERNET



## X - Window, Cliente X



Notas:

# INTERNET

---



## Clasificación de servicios

- ↳ Servicios.
- ↳ Herramientas.
- ↳ Otros.

Notas:

# INTERNET

---



## Servicios:

- ↳ Correo Electrónico ( *e-mail* ).
- ↳ Conexión remota ( *telnet* ).
- ↳ Noticias de la Red ( *usenet* ).
- ↳ Transferencia de archivos ( *ftp* ).
- ↳ Conversaciones en la red ( *talk, ytalk* ).
- ↳ Conferencia en la Red ( *Internet Realy Chat IRC* ).
- ↳ Menús de Acceso ( *gopher* ).
- ↳ Revistas Electrónicas
- ↳ Listas de correo
- ↳ Internet BBS

Notas:

# INTERNET

---



## Herramientas:

### ☐ Búsquedas:

- ↳ De personas ( *finger* ).
- ↳ De archivos ( *archie* ).
- ↳ De recursos ( *veronica & jughead* ).
- ↳ De información (servidores *wais*).
- ↳ Directorio de páginas blancas

..

## Notas:



# INTERNET

---



## Otros

- ↳ Acceso total a INTERNET ( *World Wide Web* ).
- ↳ Juegos multiusuario ( *MUD* ).

Notas:

# INTERNET

---



## Tipos de Redes de Computadoras

☐ Conmutación de circuitos.

↳ Analogía: Red telefónica .

☐ Conmutación de paquetes.

↳ Analogía: Servicio postal de correo.

Notas:

# INTERNET



## Protocolos

**T** ransmission

**I** nternet

**C** ontrol

**P** rotocol

**P** rotocol

Notas:

# INTERNET

---



## TCP/IP Arquitectura

### Protocolo a nivel de Transporte

- ↳ TCP Transmission Control Protocol.
- ↳ UDP User Datagram Protocol.
- ↳ NVP Network Voice Protocol.

Notas:

# INTERNET

---



## TCP/IP Arquitectura

### Protocolo a nivel de Red

- ↳ IP Internet Protocol .
- ↳ ICMP Internet Control Message Protocol .
- ↳ ARP Address Resolution Protocol .
- ↳ RARP Reserver Address Resolution Protocol .
- ↳ RIP Routing Information Protocol .
- ↳ EGP External Gateway Protocol .
- ↳ OSPF Open Shortes Path First.

Notas:

# INTERNET

---



## TCP/IP

**IP: INTERNET PROTOCOL**

**Brinda dos servicios básicos:**

- ↳ Enrutamiento
- ↳ Fragmentación/Re-ensamblaje

**Utiliza direcciones IP para decidir el ruteo.  
Aisla los protocolos superiores de las características  
Específicas de la Red.**

**Notas:**

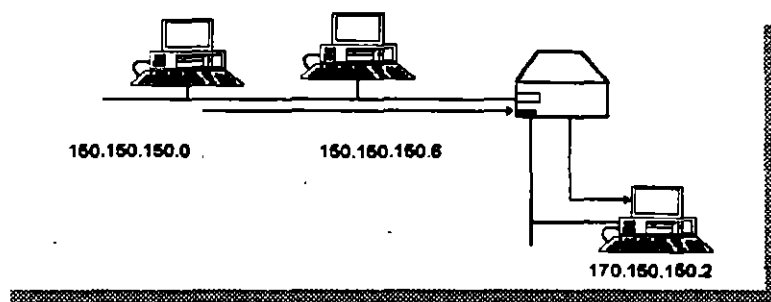
# INTERNET



## TCP/IP

IP: INTERNET PROTOCOL

ENRUTAMIENTO



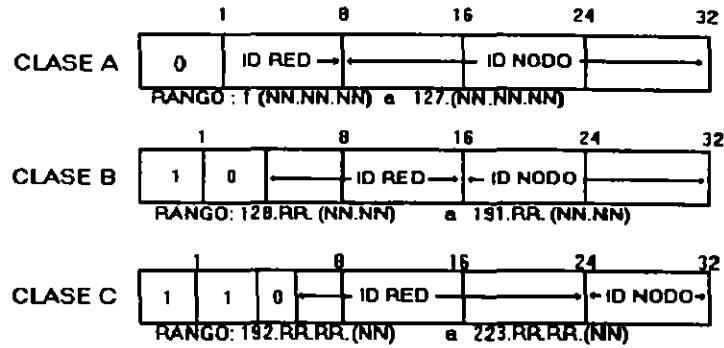
Notas:

# INTERNET



## TCP/IP

### Formato de las direcciones IP



Notas:



# INTERNET



## TCP/IP

### PROTOCOLO TCP Transmission Control Protocol

- ↳ Asignación de números de puerto para transmisión de datos
- ↳ Reconocimiento de datos recibidos
- ↳ Regulación de flujo de datos
- ↳ División de los mensajes en datagramas
- ↳ Verificación de los datagramas
- ↳ Administración
  - Establecimiento
  - Mantenimiento
  - Terminación

**Notas:**

# INTERNET

---



## TCP/IP

### PROTOCOLO NVP Network Voice Protocol

- ↻ Servicio para transporte de voz digitalizada.
- ↻ Protocolo de transacción de tiempo real.
- ↻ Utiliza IP para transmitir información.
- ↻ Emplea algoritmos de compresión.
- ↻ Es connection less.

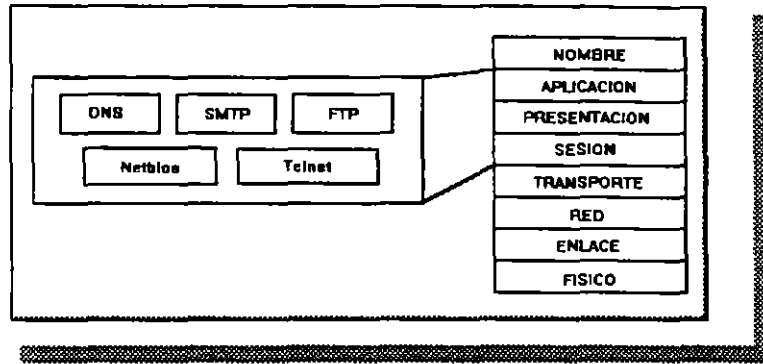
Notas:

# INTERNET



## TCP/IP

### Nivel 5-7 Sesión - Aplicación



Notas:

# INTERNET



## TCP/IP

### D N S

#### Domain Name Service

- ↳ Protocolo de nombramiento.
- ↳ Brinda traducción de nombre-dirección IP.
- ↳ Dominio: Grupo de Host.
- ↳ " Domain Name Server ".
  
- ▣ Información del servidor.
  - ↳ Dirección Internet.
  - ↳ Tipos de Computadoras.
  - ↳ Lista de servicios brindados por computadoras.
  
- ▣ Servidor
  - ↳ Servidores Maestros.
  - ↳ Primario.
  - ↳ Secundario.

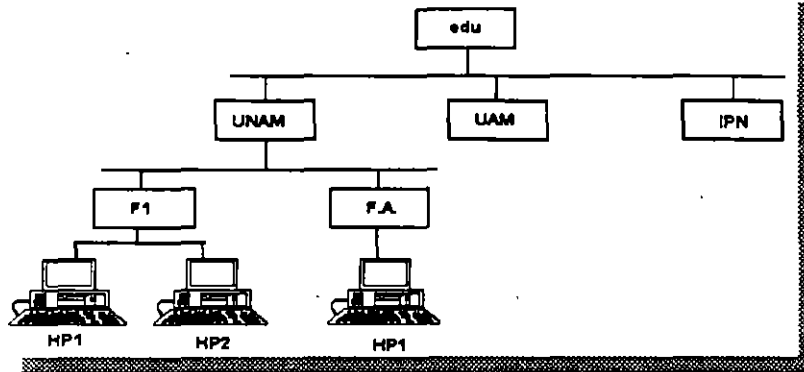
Notas:

# INTERNET



## Estructura del Sistema de Dominios

Dominios de organizaciones



Ejemplos:

Hp1.F1.UNAM.edu  
Hp1.Fa.UNAM.edu  
*gopher.gsfc.nasa.gov*

Notas:

# INTERNET



## Estructura del Sistema de Dominios

### Dominios Geográficos

Dominio	Significado.	Ejemplo:
at	Austria.	ftp.vmars.tuwien.ac.at
ca	Canadá.	debra.dgbt.doc.ca
fr	Francia.	grasp.insa.lyon.fr
mx	México.	telecom.mty.itesm.mx
uk	Reino Unido.	puffin.doc.ic.ac.uk
us	Estados Unidos.	gopher.well.sf.ca.us

Nota: En E.U. se utilizan los dominios organizacionales y en el resto del mundo la estructura geográfica. ...

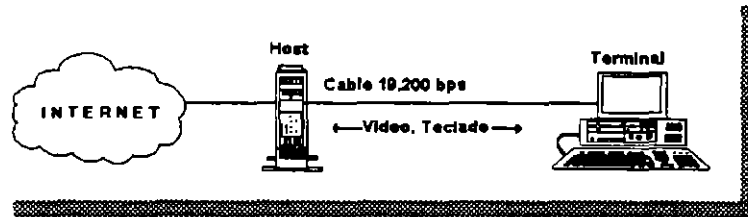
Notas:

# INTERNET



## Conexión Directa

### Tipos de Conexiones



Protocolo físico (1,2 OSI): Rs-232  
CSMA/CD

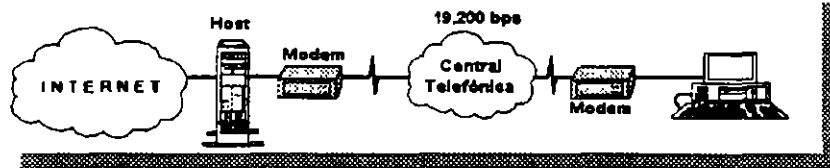
Notas:

# INTERNET



## Línea Telefónica Conmutada (Dial up)

### Tipos de Conexiones



Protocolo físico (1,2 OSI): SLIP/PPP

Notas:

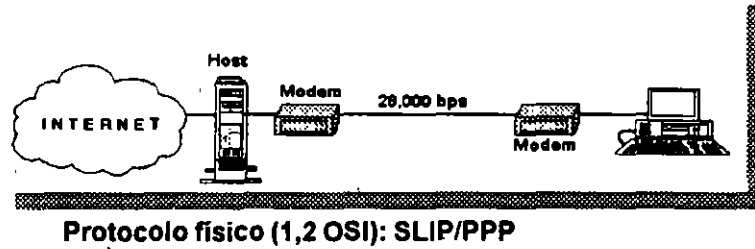


# INTERNET



## Línea Telefónica Privada

### Tipos de Conexiones



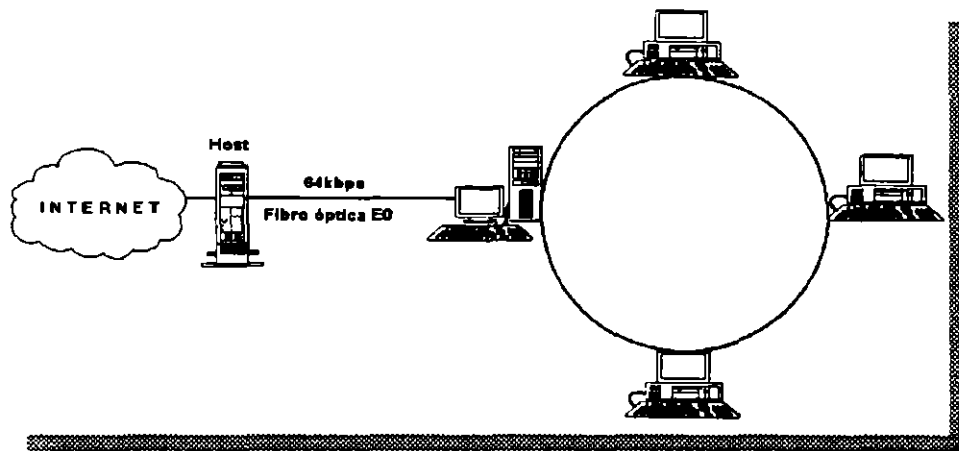
Notas:

# INTERNET



## Fibra Óptica

### Tipos de Conexiones



Protocolo físico (1,2 OSI): SLIP/PPP

Notas:

# INTERNET



## Velocidades Típicas de Acceso

Tipo de Acceso	Medio	Velocidad Máxima
Dial	Línea Conmutada	19.2 kbps
Enlace	Línea dedicada	28.8 kbps
E0	Canal E0	64 kbps
ISDNB	Línea ISNDB	128 KBPS
E1	Canal E1	2.2 mbps
T1	Canal T1	1.544 mbps
T3	Canal 3	44.746 mbps
SONET	OC -n = 1,3,12,24,48	51./2,488.2 mbps

Notas:

# CURSO: **INTERNET**

## 2.- EL CORREO ELECTRONICO Y SUS HERRAMIENTAS



Octubre de 1995.

## 2.- EL CORREO ELECTRONICO Y SUS HERRAMIENTAS

### - CORREO ELECTRONICO

#### - INTRODUCCION

El correo electrónico es sin duda, el recurso más utilizado en **INTERNET**, la mayor parte de los usuarios empiezan usando el correo electrónico, después de enviar algunos mensajes vacilantes (seguidos frecuentemente de una llamada telefónica para preguntar al destinatario si recibió el mensaje), la mayoría de los usuarios del correo electrónico se acostumbra al sistema. Después de recibir los primeros mensajes, nosotros también usaremos el correo electrónico con frecuencia y soltura, configuraremos nuestro correo para satisfacer nuestras necesidades y establecer nuestras propias listas de distribución de correo, veremos que el correo electrónico significa mucho más que cartas escritas rápidamente y memorándums, tomaremos parte en conversaciones electrónicas sobre escritores de novelas de misterio, el mercado de valores o cualquier tema de nuestro interés. En realidad el sistema de correo es un servicio general que puede transportar cualquier tipo de información: documentos, publicaciones, programas y mucho más. El único requisito es que los caracteres deben almacenarse en código ASCII, aunque bajo ciertas circunstancias es posible transmitir datos no-textuales.

Qué tan rápido nos acostumbraremos al correo electrónico, está muy relacionado con el conocimiento que tengamos del medio y con algunas decisiones técnicas básicas que se tornen al momento de elegir un sistema de correo electrónico. Existe un sinnúmero de programas de correo electrónico para cada tipo de computadora. Para obtener los conocimientos suficientes y poder describir cualquier sistema de correo electrónico, se empezará con un análisis sobre las facilidades generales de un correo electrónico, el establecimiento de domicilios y cómo funciona el correo electrónico. Después de esto, veremos cómo podemos utilizar algún programa de correo para mover archivos, tomar parte en discusiones y aprovechar algunas de las herramientas principales que existen dentro del mundo del correo electrónico.



Como cualquier otra herramienta, el correo electrónico tiene sus ventajas y sus desventajas, desde un enfoque superficial, pareciera que sólo sirve como una manera rápida de enviar cartas o algo similar, pero para saber cuándo es útil el correo electrónico, sólo hay que pensar cómo difiere de otros medios de comunicación. De alguna forma, el correo electrónico es muy similar al teléfono pero por el otro lado, es también similar al correo tradicional. La tabla 2.1 hace una comparación rápida.

Tabla 2.1: Comparación de técnicas de comunicación.

	Teléfono	Correo electrónico	Correo normal
Velocidad	Alta	Moderada	Baja
Sincronización	Sí	No	No
Formalidad	Varía	Moderada	Varía
Responsabilidad	Baja	Moderada	Alta
Facilidad de conferenciar	Grupos pequeños	Algunos a muchos	Sólo en un sentido
Seguridad	Moderada	Baja	Alta

Primero veamos qué tan rápido es posible enviar un mensaje de un punto a otro por cada uno de los medios, el teléfono ofrece una entrega inmediata y trabaja en un medio de comunicación relativamente rápido (a pesar de que su velocidad es mucho menor a la que viaja el correo electrónico). El tiempo que tarda el correo electrónico en entregar mensajes va de segundos a un día y el correo tradicional puede ser entregado de un día para otro en el mejor de los casos, pero comúnmente se toma días. El precio que hay que pagar por una comunicación telefónica rápida es la sincronización de la persona que llama y el receptor de la llamada, esto significa que ambos deben estar en el teléfono al mismo tiempo. Tanto el correo electrónico como el correo tradicional son asíncronos en el sentido que el emisor envía el mensaje cuando lo considera necesario y el receptor lo lee cuando desea. Esto se vuelve muy cómodo cuando se trata de establecer una comunicación con alguien que se encuentra a una gran distancia o cuando los horarios son muy distintos.

El tiempo que toma entregar un mensaje de correo electrónico depende del tiempo que lleva entregar el mensaje en la computadora destino y el tiempo que toma la lectura del mensaje una vez que se encuentra ahí. La primera parte es una función de cómo está conectada a la red la máquina donde se maneja su correo, y sólo lo podemos cambiar a través del tipo de conexión, y la segunda parte está bajo el control del usuario.



Si el correo electrónico no se revisa con regularidad, la entrega inmediata de los mensajes carece de sentido, el correo electrónico se hace más útil cuando se reduce el tiempo de entrega de los mensajes entre la máquina y el usuario. Es importante tratar de mantener ese tiempo menor a algunas cuantas horas. Cuando se entrega (y se lee) el correo electrónico rápidamente, puede llegar a ser tan parecido y fluído como una conversación personal.

La formalidad y la responsabilidad están íntimamente relacionadas, en el teléfono la formalidad varía: con algunas personas se es muy formal; con otras, muy informal. Lo mismo pasa con el correo tradicional, se cuenta con mucho tiempo para construir mensajes y escoger el formato que se desea (notas escritas a mano, cartas de negocios a máquina, etc.) y algunos otros datos (por ejemplo 'un sobre perfumado') dan señales tanto al emisor como al receptor acerca del propósito de la nota. El correo electrónico siempre es tecleado y no hay información en ningún libro de mecanografía acerca del formato de una carta de correo electrónico. También los individuos se pueden esconder detrás de un mensaje de correo electrónico (esto es, los mensajes del gran jefe son como los de cualquier otro usuario del correo electrónico). Como el correo electrónico fluye entre los participantes de una discusión a una velocidad que se aproxima a una conversación y como la mayor parte de la gente se siente más a gusto siendo amigable en lugar de combativo, muchos tienden a ser informales en su mensajería electrónica, pero es un problema cuando se trata de la responsabilidad: la necesidad de que los usuarios se hagan responsables de los mensajes que generan.

Los medios de comunicación escritos obligan a una mayor responsabilidad que los medios de comunicación hablados. Si se sostiene una conversación telefónica y se hace un comentario que se hubiera preferido no hacer, es posible alegar posteriormente que nunca se dijo y que la persona malentendió el comentario (confiando en que sólo una persona lo escuchó). Si se trata de hacer esto por correo, alguien puede guardar el mensaje en un archivo para posteriormente sacarlo a relucir. El único factor que reduce la posibilidad de la responsabilidad en el correo es que el autor pueda enviar un mensaje sin su identidad, como ejemplo es posible enviar un mensaje ofreciendo un puesto en la Suprema Corte con el domicilio de retorno *president@whitehouse.gov*. El correo tradicional también se puede falsificar, pero es mucho más difícil, ya que es necesario imitar sellos postales, firmas, sellos de escritorio, etcétera.



Enseguida, se necesita examinar las comunicaciones de grupo. El teléfono es un buen medio, pero sólo para grupos pequeños. Las llamadas de conferencias permiten que grupos de personas puedan hablar entre sí, pero cuando el grupo se hace más grande, tratar de programar una conferencia y el establecimiento de la misma, se vuelve muy difícil. Por otro lado, mandar correo tradicional a millones de personas es relativamente sencillo, pero el problema es que todos los mensajes se originan en un punto y van a todo el grupo, mientras que las comunicaciones desde cualquier otro punto (es decir, cualquier miembro del grupo) no pueden ser enviadas fácilmente a todo el grupo. El correo electrónico permite formar grupos de cualquier tamaño y cualquier miembro del grupo puede enviar mensajes a todos en cualquier momento. Esta cualidad del correo electrónico lo hace muy útil para diseminar información y pedir opinión a todo un grupo.

Finalmente, la seguridad del correo electrónico por lo general es deficiente comparada con los otros medios de comunicación. Si se es cuidadoso con el correo, una carta puede permanecer en un apartado postal de las oficinas del correo, hasta que llegue a manos de su destinatario. Si se abre durante el trayecto, el daño en el sobre hace evidente cualquier intrusión. Para interferir un teléfono se requiere tener acceso a la línea y contar con equipo adicional, una vez que la comunicación sale de un edificio y entra en la red telefónica, técnicamente muy difícil interferir una llamada sin la ayuda de la compañía telefónica. Sin embargo, el correo electrónico toma una ruta predecible a través de computadoras cuya seguridad puede ser cuestionable. También existen modalidades de error en las que un mensaje no se puede entregar y una computadora que no sepa qué hacer con el mensaje lo entregará al administrador del correo. Normalmente, los administradores no husmean ni divulgan el correo que les llega de esta forma, pero aun así, si la seguridad es crítica, el hecho de que un mensaje llegue a manos de cualquier persona o de un administrador responsable no es apropiado. Los "servidores de correo con seguridad mejorada" intentan cifrar los mensajes para aminorar las deficiencias en la seguridad, pero no son de uso común. Como regla general, no se puede confiar en la seguridad del correo, por lo tanto no se debe usar el correo si la seguridad es un asunto crítico.





## ▣ FUNCIONAMIENTO

El correo electrónico difiere de las otras aplicaciones porque no es un servicio de usuario a usuario, no es necesario que las máquinas emisora y receptora del correo electrónico se comuniquen directamente entre sí. Al correo electrónico se le conoce como un servicio de *almacenaje y reenvío* ya que el correo pasa de una máquina a otra hasta que llega a su destino final. Esto es análogo a la forma en la que el Servicio Postal entrega el correo. Si analizamos ésta analogía, podemos obtener algunas conclusiones interesantes.

El Servicio Postal opera como una red de almacenaje y reenvío, escribimos el domicilio en el sobre y lo depositamos en el buzón. La carta es recogida por un camión para llevarla a otro lugar y almacenara donde se clasifica y se reenvía a otro lugar, esto se repite hasta que llega a su buzón destino, no importando si el buzón destino no está en el área de cobertura del Servicio Postal, ya que en esos casos el mensaje se envía al servicio postal del país destino.

A partir de esta analogía es posible inferir un par de cosas sobre **INTERNET**. Primero, si ponemos el domicilio correctamente a un mensaje, la red se hará cargo de entregarlo, y de ser necesario, el correo puede viajar entre **INTERNET** y otras redes de correo y viceversa. El problema de hacerlo es que el domicilio se hace más complejo si se requiere que un mensaje vaya de alguna red externa a **INTERNET** y viceversa. Al igual que en el Servicio Postal, si el emisor y el receptor no están conectados en la misma red, es necesario colocar el mensaje en algún lugar en donde se concentre todo el correo que va a una determinada red. Los puntos de conexión entre redes de correo electrónico se denominan *puertas de aplicación*. Para enviar correo a través de una puerta, casi siempre es necesario proporcionar en el domicilio del mensaje información sobre cómo llegar a la puerta e información sobre cómo entregar el correo en la otra red.



Finalmente, antes de poner la carta en el buzón, la colocamos en un sobre, de la misma forma en el correo electrónico necesitamos colocar dicho sobre, el cual recibe el nombre de *encabezado del mensaje de correo*. El encabezado se refiere a lo que aparece en la primera parte del mensaje, o sea *To:*, *From:* y *Subject:*. De la misma forma que se puede modificar el domicilio de un sobre en el trayecto (por ejemplo, cuando se escribe a mano en el sobre "No enviar a este domicilio" y se pega una nota amarilla que tiene uno nuevo), así también se va agregando información al encabezado del mensaje de correo electrónico en su trayecto por la red, lo que sirve para conocer qué ruta tomó, sólo por si no llegara a su destino.

## 📧 DOMICILIOS

Si el mensaje llega o no a su destino, esto dependerá únicamente de que el domicilio haya sido construido correctamente. (A veces, el correo electrónico falla porque alguna porción de la red no funciona, pero normalmente intenta entregar todos los mensajes durante varios días, antes de abandonar) desafortunadamente, los domicilios de correo electrónico son un poco más complejos que los domicilios de las computadoras que participan en **INTERNET**, ya que en primera instancia el mundo del correo electrónico es más grande que **INTERNET**, segundo el correo electrónico se envía a una persona, no sólo a una máquina. Y tercero algunas veces los nombres de personas se incluyen como comentarios en los domicilios de correo electrónico.

Para que el correo pueda llegar a su destino ya mencionamos que es necesario colocar un domicilio correspondiente al destinatario, dentro de **INTERNET** el formato es relativamente simple, ya que por regla general contiene el nombre del usuario seguido de el caracter @ y finalmente la dirección IP del host donde se almacena el mensaje, por ejemplo: *pedro.huerta@industry.net*. Pero dado que existen otras redes de correo electrónico, para comunicarnos con ellas es necesario alterar el formato mencionado. Veamos que hacer en cada caso:



**Bitnet:** Normalmente los domicilios de Bitnet tienen la forma *nombre@equipo\_anfitrión.bitnet*. Es necesario cambiar el domicilio por *nombre%equipo\_anfitrión* y usar esto como la clave de acceso del domicilio. Se utiliza como domicilio del equipo anfitrión, el domicilio de una puerta Bitnet-INTERNET (por ejemplo, *cunyvm.cuny.edu*) y se separan ambas partes con el signo @. Por ejemplo, reescriba el domicilio Bitnet *pedro@uiucvmd.bitnet* como *pedro%uiucvmd@cunyvm.cuny.edu*. Si va a hacer esto con regularidad hay que averiguar cuál es la mejor puerta.

**CompuServe:** Los domicilios de CompuServe se componen de dos números separados por una coma. Se cambia la coma por un punto y todo se coloca del lado izquierdo del domicilio. A la derecha de la @ se escribe *compuserve.com*. Por lo tanto, el domicilio de CompuServe *76543,123* se vería de la siguiente forma *76543.123@compuserve.com*.

**Fidonet:** Los domicilios de Fidonet se componen de un nombre, un apellido y un conjunto de números con el formato *a:b/c.d*. Se separan el nombre y el apellido con un punto y se manda a *pd.fc.nb.za.fidonet.org*. Por ejemplo, para enviar un mensaje a *Murphy Shmit 1:5/2.3* se escribiría lo siguiente: *murphy.shmit@p3.f2.n5.z1.fidonet.org*. Algunas máquinas pueden tener problemas al tratar de descifrar domicilios como éstos, otra opción es tratar de hacerlo por una puerta, en el ejemplo anterior habría que tratar *murphy.shimit%p3.f2.n5.z1.fidonet.org@zeus.ieee.org*.

**Sprintmail:** Un domicilio completo en *Sprintmail* se escribe de la siguiente forma: *"Uncle Sam" /YOYODYNE/TELEMAIL/US*. Si el domicilio se utiliza dentro de *Sprintmail*, se puede simplificar a *Uncle Sam /YOYODYNE*. Estos dos parámetros se refieren al nombre de la persona y su organización, cuando alguien nos proporcione un domicilio de *Sprintmail* esto es todo lo que podremos obtener. Estos parámetros son posicionales y es necesario ponerlos en un domicilio X.500 (un "estándar" antiguo de correo) en un comando como sigue:

***/PN = Uncle.Sam/O = YOYODYNE/ADMD = TELEMAIL/C = US/@sprint.com***

Aun cuando alguna persona solamente le proporcione las primeras dos partes, se **debe** enviar el mensaje con el domicilio completo al equipo anfitrión *sprint.com*.



**MCI*mail***: Existen múltiples formas de enviar mensajes a **MCI*mail***. Los buzones de MCI requieren tanto un domicilio como el nombre de una persona asociada con él. Los libros de domicilios se parecen mucho a los directorios telefónicos. Si eso es lo que se tiene, entonces hay que utilizar el número al lado izquierdo de la @ y la puerta *mcimail.com* al lado derecho. Por ejemplo, *1234567@mcimail.com*. Si se cuenta con el nombre de la persona en **MCI*mail***, es posible enviar el mensaje con la forma *nombre\_apellido@mcimail.com*.

**UUCP**: Ahora se cambiará el domicilio UUCP, el cual tiene un formato *nombre@equipo\_anfitrión.uucp*, por el de *nombre%equipo\_anfitrión*. Hay que emplear este formato como la clave de acceso del domicilio. Después se utiliza el domicilio de una puerta UUNET-INTERNET como el nombre de la máquina, por supuesto separadas por un signo @. Por ejemplo, un usuario que recibe correo vía UUCP de PSI Inc. deberá enviar el correo a través del equipo anfitrión *uu.psi.com*, como *murphy\_s%yoyodyne@uu.psi.com*.

## AGENTE DE TRANSPORTE

### SMTP

Hay millones de personas que utilizan **INTERNET** y el sistema de correo puede transportar cualquier tipo de información que se pueda imaginar. El uso más común de este sistema permite a personas que se encuentran a una gran distancia, trabajar juntas en un proyecto común, hay muchos proyectos en los que participan personas de todo el mundo. En realidad, es muy habitual que varios individuos que no se conocen en persona y que se encuentran a miles de kilómetros de distancia, colaboren juntos en un proyecto común.

Por ejemplo, al escribir un libro, cada capítulo se puede enviar a una persona en Austria y a otra en Irlanda, para que revise el trabajo y hagan algunos comentarios. El texto de cada capítulo se puede enviar como un mensaje de correo electrónico. Una vez revisado y con los comentarios correctos introducidos en el texto, los capítulos pueden ser devueltos por correo electrónico. Gracias a la rapidez y fiabilidad de **INTERNET**, es muy común poder revisar un capítulo entero en un mismo día. El mismo proceso a través del correo tradicional puede llevar semanas.



El mismo proceso se lleva a cabo, a una mayor escala, cuando deben tomarse decisiones que afectan a **INTERNET** en sí misma. Cuando se propone una nueva idea o un nuevo estándar, todos los usuarios que lo deseen pueden participar en un debate de ámbito mundial vía correo electrónico, los grupos de discusión de *usenet* se utilizan también para este propósito.

¿Qué mantiene todo esto unido?, después de todo, **INTERNET** conecta miles de redes diferentes, cada una de las cuales tiene su propia mezcla de computadoras y software. ¿Cómo es posible que todos estos sistemas tan diferentes puedan trabajar juntos para intercambiar correo?. La respuesta es que la entrega de los mensajes está estandarizada por un sistema llamado **SMTP**, *Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo)*, que es parte de la familia de protocolos **TCP/IP**, y describe el formato de los mensajes de correo electrónico y cómo deben manipularse para realizar su entrega.

Cada computadora de **INTERNET** ejecuta un programa de correo que trabaja en segundo plano para garantizar que los mensajes son dirigidos y transportados de forma correcta. Este programa, llamado Agente de Transporte, se ejecuta en "segundo plano" (background), siempre listo para responder a cualquier petición que reciba, en la terminología Unix, estos programas reciben el nombre de demonios (daemon). Cada sistema Unix tiene varios demonios que se ejecutan ocultos en segundo plano, proporcionando de forma silenciosa servicios para nosotros, en teoría, no importa cuál sea el agente de transporte que utilice una computadora, siempre que sepa cómo enviar y recibir correo utilizando **SMTP**. La mayoría de los sistemas Unix utilizan un demonio llamado *sendmail*.

El sistema de correo electrónico funciona porque las redes de todo el mundo tienen al menos una computadora ejecutando un agente de transporte, que se encarga de enviar y recibir los mensajes, de acuerdo con el estándar **SMTP**, y en la mayoría de los sistemas, el agente de transporte es un demonio llamado *send:mail*.



## POP

La mayoría de los programas de correo para PC están basados en un sistema diferente, los hosts de **INTERNET** utilizan un protocolo llamado SMTP cuando envían y reciben correo, SMTP es parte de la familia de protocolos TCP/IP y describe el manejo de los mensajes. Para redes de PC existe otro protocolo llamado POP (Post Office Protocol). Estas computadoras pueden ejecutar un agente de usuario basado en POP que utiliza las ventajas de este protocolo.

Dentro de cada lugar de **INTERNET**, hay una o más computadoras que actúan como almacén de mensajes de correo, las cuales almacenan los mensajes que están esperando a ser leídos, con los agentes de usuario basados en POP, los mensajes no se guardan en la computadora central sino que se envían a cada una de las computadoras personales donde se almacenan hasta que son leídos.

Para entender cómo funciona esto, supongamos que se nos envía un mensaje desde el otro lado del mundo, en cada paso a lo largo del camino, el mensaje se envía de computadora en computadora sobre **INTERNET** a través de los hosts SNMP, finalmente el mensaje llega a la computadora de nuestra red que actúa como central de correos. En este punto, nuestro agente de usuario basado en POP pide a esta computadora que nos envíe el mensaje directamente a la PC, donde podremos leerlo. Los agentes de usuario basados en POP tienen importantes ventajas. Primero, dan a cada usuario un control completo sobre su correo, puesto que los mensajes se almacenan en su computadora personal. Segundo, los agentes de usuario basados en POP están programados para utilizar las características especiales de cada computadora, por ejemplo, un programa basado en DOS escrito para una PC, puede hacer uso de determinadas características especiales de la computadora, como pueden ser las teclas de función. Muchos usuarios encontrarán mucho más fácil utilizar un programa que esté especialmente diseñado para su computadora, en lugar de tener que aprender cómo utilizar, digamos, el programa estándar *mail* de Unix.

Algunos de los agentes de usuario basados en POP que se pueden encontrar son Eudora, NuPop, y Popmail/PC. Estos programas ofrecen la ventaja de utilizar nuestro entorno, sin embargo, presentan algunas desventajas importantes.



El primer problema es la seguridad. En algunas organizaciones, las PC's deben ser compartidas, por ejemplo, puede existir una sala de computadoras que puede utilizar todo el mundo. Con un programa de correo normal, nadie puede leer nuestro correo, a menos que pueda conectarse a nuestra cuenta. En otras palabras, excepto que alguien conozca nuestra palabra clave o que dejemos la terminal con la cuenta abierta nuestros mensajes están a salvo de los curiosos. Con un agente de usuario basado en POP, todos los mensajes se envían desde la computadora de correo central a nuestra computadora y se almacenan en archivos normales. Si alguien tiene acceso a la misma computadora, éste puede leer nuestros mensajes.

Aparte del problema de seguridad, hay también consideraciones logísticas. Puesto que todo el correo se envía a las computadoras personales, ¿qué ocurre si se quieren leer los mensajes desde otra computadora? Con el sistema de correo normal, es posible conectarse desde cualquier computadora de la red y acceder al correo. Con un sistema basado en POP, los mensajes que se transfieren a una computadora en particular, sólo pueden leerse en esa computadora.

Supongamos, por ejemplo, que normalmente se utilizan dos computadoras diferentes, una PC de escritorio en el trabajo y una computadora portátil en casa, con la computadora de casa usamos un módem para conectarnos con la red a través de una línea telefónica. Podemos utilizar la computadora portátil para acceder a la red y a **INTERNET** desde casa. Sin embargo, una vez que se leen en el trabajo los mensajes con un sistema basado en POP, no se podrán volver a leer estos mensajes a menos que estemos en la oficina. Algunas versiones de programas basados en POP solucionan estos problemas permitiendo leer el correo desde la computadora central sin necesidad de transferir los mensajes a nuestra computadora.

## AGENTE DE USUARIO

Como usuario, nunca trabajaremos directamente con el agente de transporte, éste funciona oculto enviando y recibiendo correo vía SMTP. El programa que utilizaremos se llama Agente de Usuario. El agente de usuario actúa como interfaz con el sistema de correo de **INTERNET**, permitiendo leer los mensajes, redactar mensajes nuevos, eliminar mensajes que ya se han leído y mucho más.



En **INTERNET** se utilizan muchos agentes de usuario diferentes. Existen para sistemas Unix y para sistemas PC y Macintosh, además existen en modo texto o para ambientes gráficos, y más adelante describiremos algunos detalles de los más comunes. El agente de usuario que utilice para leer y redactar los mensajes dependerá de cuál esté disponible en el sistema. En algunos sistemas, sólo hay un programa de correo. En otros sistemas, se puede elegir entre varios.

No obstante la gran variedad, el agente de usuario más extendido es el programa *mail* de Unix, el cual forma parte de las utilerías de cualquier sistema Unix y por tanto, está disponible para la mayoría de los usuarios de **INTERNET**.

## AGENTES DE USUARIO MAS COMUNES

### AGENTES DE USUARIO PARA SISTEMAS UNIX

#### *mail*

El *mail* de Unix es la base de todos los paquetes de correo electrónico, es agente de usuario más común dentro de **INTERNET** y además se encuentra incluido en las utilerías de cualquier sistema operativo Unix. La funcionalidad de este agente es limitada comparada con otros agentes comerciales, pero la garantía es que cualquier agente basado en SMTP es cubierto por *mail*.

Existen dos formas de ejecutar el programa *mail*, la primera es escribir la orden seguida de la dirección de correo a la cual se desea enviar nuestro mensaje, y la otra es digitar solamente la orden *mail* y aparecerá nuestro buzón, del que podremos seleccionar algún mensaje para leerlo, responderlo, borrarlo, etc.

Mientras se está escribiendo un mensaje, existen algunas funciones especiales que podemos ejecutar a través de las instrucciones de escape tilde (~), a continuación mostramos un resumen:





Tabla 2-2 Resumen de las órdenes de escape Tilde de *Mail*.

<i>Orden</i>	<i>Descripción</i>
<i>~?</i>	<i>Ayuda: muestra un resumen de las órdenes de escape tilde</i>
<i>~b dirección</i>	<i>Añade una dirección a la línea "Blind copy"</i>
<i>~e dirección</i>	<i>Añade una dirección a la línea "Copy"</i>
<i>~d</i>	<i>Lee el contenido del archivo dead.letter</i>
<i>~e</i>	<i>Invoca el editor de texto</i>
<i>~f mensajes</i>	<i>(continua) incluye un mensaje</i>
<i>~h</i>	<i>edita todas las líneas de la cabecera</i>
<i>~m mensajes</i>	<i>incluye un mensaje, desplazado a la derecha una tabulación</i>
<i>~p</i>	<i>muestra el mensaje actual</i>
<i>~q</i>	<i>termina (igual que pulsar CTRL-C dos veces)</i>
<i>~r archivo</i>	<i>incluye el contenido de un archivo.</i>
<i>~s asunto</i>	<i>cambia la línea de "Subject"</i>
<i>~t dirección</i>	<i>añade una dirección a la línea "To"</i>
<i>~y</i>	<i>invoca un editor alternativo</i>
<i>~w archivo</i>	<i>escribe el mensaje actual en un archivo</i>
<i>~! orden</i>	<i>ejecuta una orden del sistema y vuelve al mensaje</i>
<i>~  orden</i>	<i>pasa el mensaje actual por un filtro.</i>

Existen diversas alternativas dentro del correo electrónico, podemos enviar copias de nuestros mensajes, contestarlos, mandar una copia oculta, saber si un mensaje ha sido leído, comprobar la recepción de correo, etc. Es recomendable revisar la documentación correspondiente a nuestro sistema.

Enseguida mostramos un resumen de comandos de *mail*.



Tabla 2-3. Resumen de las órdenes mail más importantes

<i>Abreviatura</i>	<i>Nombre</i>	<i>Descripción</i>
<b>Terminar mail</b>		
<i>q</i>	<i>quit</i>	<i>Salida de mail</i>
<i>x</i>	<i>exit</i>	<i>Salida de mail, abandonando todos los cambios</i>
<b>Ayuda</b>		
<i>?</i>	<i>-</i>	<i>Presenta un resumen de órdenes</i>
<i>!</i>	<i>list</i>	<i>Lista los nombres de todas las órdenes disponibles</i>
<b>Cabecera</b>		
<i>sh</i>	<i>headers</i>	<i>Muestra resúmenes de cabecera</i>
<i>z</i>	<i>-</i>	<i>Presenta la siguiente serie de resúmenes de cabecera</i>
<i>z-</i>	<i>-</i>	<i>Presenta la serie anterior de resúmenes de cabecera</i>
<b>Mostrar Mensajes</b>		
<i>+</i>	<i>-</i>	<i>Muestra el mensaje siguiente</i>
<i>-</i>	<i>-</i>	<i>Muestra el mensaje anterior</i>
<i>RETURN</i>	<i>-</i>	<i>Muestra el mensaje siguiente</i>
<i>número</i>	<i>-</i>	<i>Muestra el mensaje #número</i>
<i>n</i>	<i>next</i>	<i>Muestra el mensaje siguiente</i>
<i>p</i>	<i>print</i>	<i>Muestra mensajes (en pantalla)</i>
<i>to</i>	<i>top</i>	<i>Muestra las primeras líneas de un mensaje</i>
<b>Contestar y Enviar</b>		
<i>m</i>	<i>mail</i>	<i>Envía un mensaje nuevo al usuario indicado</i>
<i>r</i>	<i>reply</i>	<i>Contesta al remitente y a todos los receptores</i>
<i>R</i>	<i>Reply</i>	<i>Contesta sólo al remitente</i>
<b>Procesar un Mensaje</b>		
<i>d</i>	<i>delete</i>	<i>Borra un mensaje</i>
<i>dp</i>	<i>-</i>	<i>Borra el mensaje actual, muestra el siguiente mensaje</i>
<i>e</i>	<i>edit</i>	<i>Ejecuta el editor de textos para el mensaje</i>
<i>pre</i>	<i>preserve</i>	<i>Mantiene mensajes en el buzón del sistema</i>
<i>s</i>	<i>save</i>	<i>Guarda mensajes en el archivo especificado</i>
<i>u</i>	<i>undelete</i>	<i>Recupera los mensajes previamente borrados</i>
<i>w</i>	<i>write</i>	<i>Lo mismo que <i>s</i>, pero no guarda la cabecera</i>
<b>Ordenes Shell</b>		
<i>!</i>	<i>-</i>	<i>Ejecuta una orden de shell</i>
<i>sh</i>	<i>shell</i>	<i>Interrumpe mail, comienza un nuevo shell</i>



La primera versión de *mail* fue muy primitiva, ofreciendo sólo las funciones más rudimentarias para redactar y leer mensajes. Hace algunos años, aparecieron versiones nuevas de *mail* más desarrolladas y mucho más potentes, al principio, había dos versiones de este programa, una para cada uno de los principales sistemas Unix: Unix de Berkeley (conocido como BSD, por Berkeley Software Distribution) y Unix System V (el cual viene de AT&T Bell Labs). El agente de usuario de BSD se denominó *mailx*. El agente de usuario del Sistem V se llamó *Mail*. (Unix distingue entre letras mayúsculas y minúsculas, por tanto, *Mail* y *mail* son dos nombres diferentes. los usuarios tenían que acordarse de utilizar el nombre correcto, si usaban *mailx* (o *Mail*), querían utilizar la versión nueva del programa de correo. Si por el contrario utilizaban *mail* -el nombre más lógico- deseaban usar la versión antigua. Actualmente, el programa viejo ha caído en desuso. Cuando se introduzca la orden *mail*, automáticamente se dispondrá de *mailx* o *Mail*.

### ↳ Elm

Elm es un programa de pantalla completa que es muy sencillo de utilizar y fácil de aprender, soporta algunas funciones de MIME (Extensiones Multipropósito del Correo INTERNET), y quizá su característica más importante sea la facilidad que tiene para ser configurado a gusto del usuario.

### ↳ Pine

Pine utiliza un sistema de menús que satisface especialmente a los principiantes o usuarios ocasionales, Pine es magnífico para personas inexpertas que quieran sentarse en una computadora y comenzar a utilizar el correo inmediatamente, las opciones de cada uno de los menús son muy claras y fáciles de entender y permiten consultar información de ayuda en cualquier momento. Sin embargo, si se envían o reciben muchos mensajes, será mejor utilizar un agente de usuario diferente. Aunque Pine es muy fácil de aprender, los otros programas son más potentes y elegantes además de más apropiados para usuarios con alguna experiencia. Lo que mantiene a Pine dentro de la competencia es la facilidad para utilizar MIME.



## ↳ MH

Otro agente de usuario que se puede encontrar se llama MH (Message Handler). MH se compone de una serie relativamente simple de programas de propósito único, en lugar de utilizar un sólo programa para manejar todo el correo, dispone de programas diferentes para cada tarea. Cuando se utiliza *mail* o Elm, se teclea una orden para arrancar programa, cada tarea que se quiera ejecutar se realiza con un mismo programa. Con MH, se utilizan órdenes distintas para ejecutar cada tarea. Por ejemplo, para comprobar qué mensajes nuevos se han recibido se utiliza la orden *scan*; para ver los mensajes, se utiliza la orden *show*; para contestar un mensaje se utiliza la orden *reply*; para enviar un mensaje nuevo, se utiliza la orden *comp*. El principal inconveniente de este sistema es que se necesita memorizar muchas órdenes, pero su principal ventaja es que no necesitamos terminar lo que estamos haciendo para trabajar con el correo. Se puede dar una orden MH en cualquier momento para ejecutar una sola tarea. Por ejemplo, si se necesita examinar un documento, con *mail* o con Elm necesitaríamos salir del programa (suspenderlo momentáneamente) para ejecutar la orden, mientras que con MH como no se trabaja dentro de un programa de correo, simplemente se teclea la orden, se comprueba la información y después se utiliza una orden de MH para enviar la contestación.

## ↳ Mush

Otro agente de usuario que se puede encontrar en los sistemas es Mush, el nombre corresponde a **M**ail **U**ser's **S**hell. Mush es flexible y se puede utilizar de dos formas: como *mail* (utiliza una interfaz de órdenes) o como Elm (utilizando una interfaz de pantalla completa). Existe también un producto comercial, llamado Zmail, que está basado en Mush.

## ↳ Otros Agentes

Para aprovechar las características de la interfaz gráfica de X-Window, para los usuarios de Sun, existe *Mailtool*, y para los usuarios de Emacs existe *Rmail*.



## AGENTES PARA PC'S

### ↳ Pegasus

También llamado P-Mail, es un agente de usuario para redes de área local, no soporta POP pero trabaja como el MH de Unix y soporta SMTP.

### ↳ Pine

Proporciona las mismas características funcionales que su versión para sistemas Unix, con la salvedad de que se ejecuta bajo DOS. No soporta POP.

### ↳ Chameleon

Fabricado por NetManage es una de las mejores y más completas implementaciones de TCP/IP para Windows, incluye servicios de SLIP y PPP, soporta a Winsock e incluye servicios de *ftp*, *telnet*, *Mail*, *Ping*, *finger*, *Whois*, *Gopher*, *News*, soporte a MIME y a agentes SNMP.

### ↳ CommSet

Diseñado por Cybernetic Control INC, incluye servicios como *Mail*, un lector de noticias, *finger*, *ftp*, *telnet*, *gopher* y *whois*.

### ↳ Eudora

Ciente Windows de correo electrónico de fácil acceso, soporte a POP3 (versión mejorada de POP), la versión 1.XX es gratuita mientras que las versiones 2.XX y posteriores son comerciales.

### ↳ ECStmail

Sin duda uno de los más versátiles, soporta MIME y existen versiones servidor para Unix, OS/2, Open VMS y Windows NT, mientras que los clientes pueden ejecutarse bajo DOS, Windows y Mac System 7.

### ↳ iGate

De Smart Systems es un agente SMTP para WordPerfect Office, corre bajo Office Connection Server y no necesita TCP/IP ni MHS.



## ↳ Iride

Es un agente MIME para Macintosh, desarrollado por GNBTS (Gruppo Nazionale Bioingegneria sezione di Trieste). Soporta POP y SMTP y corre bajo MacTCP 1.1 o posterior.

## ↳ Mail-it

De Uniplam Ltd, es un cliente de correo para Windows 3.1, ofrece integración total con MAPI (Mail API), soporta SMTP, POP, UUCP y MIME. Dentro de sus cualidades destacan la utilización de folders para organizar el correo, un libro de direcciones y la interacción con aplicaciones mail-enabled y con servicio de directorios MAPI.

## ☰ ENVIO DE DATOS BINARIOS POR CORREO

Algunas veces, se requiere enviar un archivo en formato binario (por ejemplo, un archivo de WordPerfect, respaldos de discos, etc.) a través del correo electrónico. El programa *ftp* (que veremos más adelante), originalmente fue diseñado para la transferencia de archivos, pero puede no ser práctico o capaz, ya que un mensaje de correo electrónico puede llegar a donde *ftp* no puede y pasar a través de redes que no están directamente conectadas a **INTERNET**, o a través de redes que sólo proveen el servicio de correo electrónico. Además, *ftp* no puede enviar un archivo a muchos destinatarios; es posible que se desee enviar el archivo ejecutable de un nuevo lector de News a una enorme lista de distribución de correo, en cuyo caso *ftp* sería poco práctico.

Debido a esto, el correo electrónico proporciona ciertas ventajas, pero desafortunadamente, es un medio de comunicación sólo para texto, es decir, que únicamente se pueden manejar mensajes que fueron construidos con caracteres. Como se mencionó antes, cualquier sistema de correo puede permitir la transferencia de archivos de texto en un mensaje, sin embargo, sólo algunos permiten la transferencia en forma directa de archivos binarios (que no son de texto), y aquellos que son capaces de hacerlo seguramente no son compatibles entre sí, aunque MIME está estableciendo estándares en esta área.



## UUENCODE/UUDECODE

Con un poco de trabajo adicional, cualquier sistema de correo puede transferir un archivo binario, siempre teniendo en cuenta que tanto el emisor como el receptor utilizan el mismo esquema de conversión de binario a ASCII. Todos los sistemas UNIX soportan esquemas estándares a través de las mismas utilerías, una de ellas es llamada *uuencode*, la cual se utilizará como ejemplo. Muchos otros sistemas tienen una utilería equivalente.

Por supuesto, la habilidad de transferir un archivo en binario no significa que le será útil al que lo reciba. Es posible enviar a alguien un archivo tipo *tar* que contenga el respaldo de un directorio, pero sólo será útil si los destinatarios cuentan con una utilería para restaurar los archivos contenidos, si esos archivos son ejecutables, sólo serán útiles si el emisor y el receptor tienen el mismo tipo de computadora. Sin embargo, si el archivo binario que se envíe es útil o no, nada tiene que ver con el correo electrónico.

No importa la velocidad, si se cuenta con las mismas utilerías en los sistemas de ambos extremos todo está resuelto. Se necesita encontrar un programa que convierta un archivo binario en una representación de caracteres que pueda ser impresa, (Nosotros mismos podríamos hacer este software tomando cada byte y convirtiéndolo en un número entre 0 y 255, para después almacenar esta secuencia de números de tres dígitos en otro archivo). Y se necesita otro programa que realice la función inversa, para que la representación sea reconvertida a binario. La mayoría de los sistemas tiene esta utilería disponible, el programa *uuencode* viene incluido en UNIX, *btoc* está disponible en muchos sistemas UNIX y ambos están disponibles al público en general para utilizarlos en PC. El programa *BinHex* realiza la misma función en el mundo de las Macintosh. Con estas utilerías es posible cifrar el archivo, convirtiendo el archivo binario en una representación de texto, una vez que se tiene una representación de texto, es posible enviarla a través del correo electrónico. El receptor toma el mensaje, elimina los encabezados (algunas veces esto no es necesario), lo descifra y de nuevo se tiene el archivo binario original. He aquí un ejemplo:



```
% uuencode libro1a < libro1a > temp
% mail pedro.huerta@isa.industry.net
Subject: Programa
Aqui esta el programa que querias (necesitas uudecode):
~r temp
"temp" 7216/447224
.
Cc:
%
```

La primera línea (*uuencode*) toma el archivo "libro1" y lo cifra en un archivo que es posible imprimir, pone el resultado en el archivo "temp". En el proceso, internamente pone al archivo el nombre "libro1", enseguida, se redacta un mensaje de correo electrónico dirigido a *pedro.huerta@isa.industry.net*, proporcionándole ayuda sobre qué hacer con el archivo, la línea *~r* es el comando del sistema de correo para copiar el archivo en el mensaje, aquí se copia el archivo "temp" creado con anterioridad. La siguiente línea dice que el archivo "temp" fue insertado y que tiene 7216 líneas y 447 224 caracteres de longitud. (algunos sistemas de correo tienen límites para el tamaño de los mensajes; por ejemplo 1,000 líneas o 50,000 caracteres. Si se rebasan estos límites es posible emplear la utilería *split* de Unix para separar el archivo en partes y enviarlas por correo individualmente). Teclear un punto (.) en la columna 1 indica que es el "fin del mensaje" (eot "end of transmission"). Un retorno como respuesta al despliegue de *Cc:* indica que no hay "ninguna copia al carbón para nadie".

Más tarde, cuando Pedro se conecte para revisar su correo, verá lo siguiente:





```

% mail
"/usr/spool/mail/pedro": 1 message 1 unread
> N 1 saul Sat Nov 2 19:57 7245/448458 "Program"
& 1
Received: from decfi.fi.unam.mx by isa.partners.industry.net with. . .
(5.65c/IDA-1.4.1 for <pedro.huerta@industry.net>;
Sat, 2 Nov 1991 19:56:31 -0600
Received: by decfi.fi.unam.mx id AA22546
(5.65c/IDA-1.4.4 for pedro.huerta@industry.net);
Sat, 2 Nov 1991 19:56:16 -0600
Date: Sat, 2 Nov 1991 19:56:16 -0600
From: Saul Magana <smagana@decfi.fi.unam.mx>
Message-Id: <199111030156.AA22546@decfi.fi.unam.mx>
To: pedro.huerta@industry.net
Subject: Programa

Aquí esta el programa que querias (necesitas uudecode):
begin 755 overthrustrer
M'.....!@OD.....!O !< @"!J< '4H'
M+ .....%!"$.....!!!!!!
.....

CTRL-C Interrupt
& s progcod
"progcod" [New file] 7244/448458
& q
% uudecode progcod

```

Pedro ve uno de los mensajes de "saul" que ha sido el último en llegar y que no ha sido leído, el cual es enorme (7244 líneas). En respuesta al indicador &, introduce el comando 1, que significa "leer el mensaje 1". Todo empieza como un mensaje normal, hasta que se topa con una línea que empieza con la palabra *begin*; éste es el principio de un archivo binario codificado. Después de ver un par de líneas, interrumpe el listado (tecleando CTRL-C) y después teclea el comando *s progcod*, con lo que guarda el mensaje en un archivo llamado "progcod", después, pedro ejecuta el programa *uudecode* en el nuevo archivo. El programa ignora el texto del principio hasta llegar a la línea que contiene la palabra *begin* y a continuación descifra el código ASCII.



Al hacer esto, crea un archivo binario igual al original, el archivo se queda con el nombre original que se le asignó dentro del comando *uuencode* (libro1), el cual fue grabado en la línea "begin". Al final del proceso pedro cuenta con el programa "libro1".

## ☞ MIME

MIME (Multi-purpose INTERNET Mail Extensions, 'Extensiones Multipropósito del Correo INTERNET') es una especificación para enviar objetos que no sean mensajes de texto en forma automática mediante el uso del correo electrónico. Esto permite evitar lo que se mencionó anteriormente acerca de *uuencode* y *uudecode*. Aun cuando nosotros y el receptor del mensaje estemos empleando distintos paquetes de correo, si los dos paquetes cumplen con la especificación MIME, deberá ser posible enviar cualquier tipo de objetos, dentro de lo razonable.

Aunque MIME es fundamentalmente una especificación para anexar archivos en mensajes de correo electrónico, la mayoría de la gente asocia a MIME con multimedia, por lo tanto sus expectativas van un poco más lejos. Los sistemas de correo MIME saben cómo procesar diversos tipos de archivos anexos, en particular imágenes, grabaciones de audio y películas. (Poder desplegar este tipo de archivos dependerá de que la computadora receptora cuente con el hardware apropiado y con algún software adicional.)

Gracias a esto es posible enviar un mensaje en donde se describan las características de un nuevo juguete a control remoto que se esté intentando lanzar al mercado, anexando un video donde el juguete aparezca persiguiendo a un perro en una sala. El receptor podrá ver el mensaje y sin tener que hacer nada más, ver el video. En su máximo nivel de sofisticación, algunos mensajes MIME pueden llegar a contener software para ser ejecutado en su computadora. Por ejemplo, podríamos recibir un mensaje que describa las condiciones de los orfanatorios ciudad de México, y después de leer este mensaje, se crea automáticamente una nueva pantalla que despliega un modelo de la "carta para su representante consejero ciudadano" que dice "si se desea enviar esta carta vía correo electrónico, diríjase a ..." y la carta es redactada, firmada y enviada automáticamente.



Aunque el hecho de permitir a alguien que ejecute software en su computadora suene peligroso, existe aquí una excelente pieza de ingeniería: este software está escrito en un lenguaje especial para que únicamente sea entendido por su sistema de correo MIME, y éste no permite que alguien haga cosas que pudieran causar daño.

Todas estas funciones tienen tanto su lado positivo como su lado negativo, contenidos en el mismo paquete. El lado positivo consiste en que permiten a personas con poco conocimiento técnico la posibilidad de enviar todo tipo de mensajes como parte del correo electrónico. El lado negativo es que estas personas no saben que no todas las funciones son universales y que algunos receptores no podrán descifrar sus mensajes.

Los sistemas de correo con MIME saben cómo descifrar diferentes tipos de archivos, sin necesidad de que el usuario tenga que realizar algún descifrado aparte. Sin embargo, todavía se presenta un problema de compatibilidad ya que existen varias clases de MIME. Un paquete que cumple con la especificación MIME debe ser capaz de manejar mensajes en formato MIME pero no está obligado a manejar todos los posibles tipos de mensajes. En general, los paquetes de correo electrónico tanto del emisor como del receptor, definen un conjunto de tipos de MIME que conocen; en algunos casos coincidirán pero en otros no.

Otra característica de MIME es que puede evitar enviar un mensaje a menos que el destinatario lo acepte. En lugar de poner el archivo en una porción del mensaje como una inserción codificada, le agrega una directiva diciendo al sistema de correo del receptor cómo hacer un *ftp* en forma automática. (Por supuesto que el emisor debe asegurarse de que el archivo está disponible en algún servidor de *ftp*.) Esta característica permite decidir si se quiere o no recibir el archivo; si no se desea el archivo, se ahorra espacio de almacenamiento en su máquina y se reduce el tráfico en la red. Por ejemplo, es posible enviar un mensaje de correo electrónico con el borrador de un nuevo manual del usuario para todos los empleados de una compañía. Realmente el manual del usuario no se enviará a todos los usuarios; simplemente estará depositado en un servidor *ftp* esperando a ser copiado por quien lo necesite. Aquellos que lo requieran lo pueden recibir sin siquiera saber cómo usar el *ftp*, el sistema de correo MIME se encargará de realizar la transferencia vía *ftp*. Aquellos que no requieran del manual, pueden ignorarlo.



## FINGER

### INTRODUCCION

Con tantos millones de personas utilizando **INTERNET**, es muy importante poder encontrar información sobre una persona en particular. El servicio *finger* permite hacer precisamente esto, ya que con una simple orden, se puede encontrar información pública sobre otro usuario de **INTERNET**.

El servicio *finger* es un sistema cliente/servidor que suministra tres tipos principales de información, primero, se puede ver determinada información pública sobre cualquier usuario de un host de **INTERNET**. Dicha información puede variar de un host a otro, y algunos hosts no mostrarán ninguna información por razones de seguridad. En general, no obstante, la mayoría de los hosts suministran información básica sobre sus usuarios.

Todo lo que se necesita saber es el host que utiliza una persona y su identificador de usuario, nombre o apellido. Con esta información, se puede utilizar el servicio *finger* para consultar toda o parte de la siguiente información:

- ◇ El identificador de usuario de la persona
- ◇ Su nombre completo
- ◇ Si el usuario está conectado
- ◇ La fecha de la última vez que el usuario utilizó la cuenta
- ◇ Si ha leído o no su correo
- ◇ Un número de teléfono
- ◇ La dirección de la oficina
- ◇ Información que dicho usuario ha preparado específicamente para su publicación (por ejemplo, las horas de oficina)

Una segunda forma de hacer uso del servicio *finger* es comprobar si un usuario está utilizando actualmente un host de **INTERNET**. Se puede ver un resumen que nos dice todo o parte de la siguiente información para cada usuario que está conectado:



- ◇ Identificador de usuario
- ◇ Nombre completo
- ◇ Cuándo se conectó
- ◇ Cuánto tiempo ha pasado desde que se pulsó la última tecla
- ◇ Número de teléfono y ubicación de la oficina
- ◇ Nombre de la computadora o servidor de terminales desde el que ese usuario se conectó

La tercera forma de utilizar *finger* es conectarnos con determinados hosts de **INTERNET** que se han configurado para ofrecer otros tipos de información. Por ejemplo, un host puede suministrar información sobre los últimos terremotos.

## CONSULTA DE INFORMACION

Para ver información sobre una persona, es necesario conocer la dirección de su computadora, su identificador de usuario, o su nombre o apellido. La orden *finger* tiene el siguiente formato:

*finger nombre@dirección*

Por ejemplo, supongamos que se quiere ver información sobre una persona llamada John Smith que trabaja en en la Compañía Acerera Nacional. Su identificador de usuario es *jsmith* y la dirección de su computadora es *acerera.com.mx*. lo que debemos hacer es teclear:

*finger jsmith@acerera.com.mx*

Si no conocemos el identificador de usuario podemos intentar con las siguientes variantes:

*finger john@acerera.com.mx*

*finger smith@acerera.com.mx*

Puede darse el caso que existan varias personas con el nombre smith, y en su caso *finger* mostrará información sobre todos los usuarios con ese nombre, por el contrario si un servidor *finger* no es capaz de localizar a la persona que se le ha indicado, responderá con un mensaje que no contiene ninguna información.



Además de la información que ya mencionamos, *finger* muestra el contenido de dos archivos, *.project* y *.plan*, que deben estar en el directorio inicial de cada usuario. Si el archivo *.project* existe, *finger* mostrará la primera línea del archivo. Si existe el archivo *.plan*, *finger* mostrará todo su contenido. Y para crearlos solamente es necesario utilizar algún editor de texto como el *vi* de Unix.

Antes que nada, es necesario entender que la información suministrada por una orden *finger* depende del servidor remoto de *finger*. Por eso, se obtendrán diferentes tipos de información dependiendo del host al que se haga *finger*. Algunos servidores de *finger* están configurados para mostrar información sobre el correo y otros simplemente no lo harán.

El correo de cada usuario se almacena en un archivo propio llamado un buzón, y cuando se hace *finger* a un usuario, el servidor de *finger* puede examinar el buzón del usuario. El servidor comprueba la última vez que se accedió al buzón y la compara con la última vez que llegó correo nuevo, y en base a esta comparación *finger* nos dará diferentes mensajes de salida, nos indicará si el buzón está vacío, si no se ha accedido al buzón desde que llegó el correo, si se ha accedido desde que llegó el correo o si no ha llegado correo desde la última vez que accedieron al buzón.

El problema es que es posible acceder al buzón sin leer ningún mensaje, lo cual lleva a que los mensajes de *finger* pueden ser engañosos.

## OTROS SERVICIOS DE FINGER

Hasta ahora, en todos los ejemplos se ha especificado el nombre y la dirección de un usuario en particular y *finger* responde a esto mostrando información sobre dicho usuario. Otra forma de utilizar *finger* es especificar sólo la dirección, con lo que el servidor *finger* en el host remoto responderá con una lista de todos los usuarios que hay conectados.

El formato de esta orden es:

*finger @dirección*



*finger* tiene algunas limitaciones que debemos analizar. Primero, no todos los hosts de **INTERNET** suministran un servicio *finger*. En la mayoría de las ocasiones, este servicio sólo será proporcionado por las máquinas Unix, pero debemos considerar que hay muchas computadoras no-Unix en **INTERNET**. Aunque hay algunos hosts no-Unix que tienen servicio *finger*, esto no es lo común. Si se hace *finger* a un host y no se recibe respuesta, es posible que se haya encontrado una computadora no-Unix, como una PC o Macintosh, o una VAX ejecutando VMS.

Segundo, hay muchos hosts Unix que no contestarán a las peticiones de *finger* realizadas fuera de su red local, por razones de seguridad o simplemente para disminuir el tráfico de la red. La segunda limitación es que no todos los servidores remotos que proporcionan servicio *finger* mostrarán toda la información de la que disponen. Por ejemplo, algunos hosts no mostrarán el contenido de los archivos *.project* y *.plan* a los usuarios foráneos.

Finalmente en algunos hosts de **INTERNET** se utiliza el servicio *finger* para suministrar otra clase de información especializada, basta con revisar un catálogo de recursos para saber quienes nos proporcionan otro tipo de información a través de *finger*, como ejemplo podemos mencionar al servicio de resultados de los partidos de béisbol al cual accesamos con la orden:

*finger jtchern@sandstorm.berkeley.edu*



# INTERNET



---

## Correo Electrónico

- ↳ Recurso más utilizado en Internet.
- ↳ Columna vertebral de Internet.
- ↳ Intercambia información entre usuarios Internet.

Notas:



# INTERNET



---

## Correo Electrónico

### ☐ Tipos de Información.

- ↳ Mensajes.
- ↳ Documentos.
- ↳ Publicaciones.
- ↳ Programas.
- ↳ Archivos en general.

Los archivos que viajan deben ser tipo texto.

Notas:

# INTERNET



---

## Correo Electrónico

### ☐ Características.

- ↻ Asíncrono (No en línea).
- ↻ Poco formal.
- ↻ Responsabilidad moderada.
- ↻ Se puede establecer comunicación en grupos.
- ↻ Debe revisarse con regularidad.
- ↻ Seguridad baja.

Notas:

# INTERNET



---

## Correo Electrónico

- ↳ Su funcionamiento se basa en un servicio de *almacenaje y reenvío*.
  
- ↳ Analogía con el servicio postal.
  
- ↳ Se puede utilizar entre Redes diferentes a Internet.

**Notas:**

# INTERNET



---

## Correo Electrónico

### Formatos de domicilios

- ↪ Internet:            `pedro@industry.net`  
formato :            `nombre@host`
  
- ↪ Bitnet:             `nombre@host.bitnet`  
en internet:        `nombre%host@puerta bitnet-internet`
  
- ↪ CompuServe:       `12847,530`  
en internet:        `12847.530@compuserve.com`
  
- ↪ uucp:                `nombre@host.uucp`  
en internet:        `nombre%host@puerta uucp-internet`

Notas:

# INTERNET



---

## Correo Electrónico

### ☐ Protocolos (Agentes de transporte)

↳ **SMTP** (Pertenece a TCP/IP).  
Simple Mail Transfer Protocol.

↳ **POP** (para pc's).  
Post Office Protocol.

**Notas:**

# INTERNET



---

## Correo Electrónico

### ☐ Agentes de usuario

- ↳ *mail* (de Unix)
- ↳ Mail (versión nueva de Unix).
- ↳ Elm.
- ↳ Pine.
- ↳ MH.
- ↳ Mush
- ↳ Mailtool (Sun).

**Notas:**

# INTERNET



---

## Correo Electrónico

### ☐ Agentes de usuario para Pc's

☞ Pegasus

☞ Pine

☞ Chameleon

☞ Eudora

☞ ECSmail

☞ iGate

Notas:

# INTERNET



---

## Correo Electrónico

### Envío de datos Binarios

↳ MIME (Multipurpose Internet Mail Extensions ).

↳ UUENCODE (Binario → Texto)

↳ UUDECODE (Texto → Binario)

Notas:



# INTERNET



---

## Correo Electrónico

### ☐ Servicio *Finger*

↳ Encuentra información pública sobre los usuarios de Internet.

↳ Arquitectura cliente/servidor.

Notas:

# INTERNET



---

## Correo Electrónico

### ☐ Servicio Finger

Proporciona 3 tipos principales de información:

- ☞ Datos particulares de un usuario.
- ☞ Si el usuario está utilizando un Host de Internet, en un momento determinado.
- ☞ Servidores finger con información específica (p.e. últimos terremotos)

**Notas:**

# INTERNET



---

## Busquedas de personas

- ↳ Finger
- ↳ Directorio de páginas blancas
- ↳ Servidores Whois
- ↳ Fred y los directorios X.500
- ↳ Netfind

Notas:

# INTERNET



---

## Servicio *finger*

↳ Formato:

`finger nombre@dirección`

↳ Ejemplo:

`finger Rserret@cce.com.mx`

`finger rafa@cce.com.mx`

`finger serret@cce.com.mx`

Notas:

# CURSO: **INTERNET**

## 3.- GRUPOS DE NOTICIAS



Octubre de 1995.

### 3.- GRUPOS DE NOTICIAS

#### **Usenet**

##### INTRODUCCION A *Usenet*

*Usenet* es una gran colección de grupos de discusión en los que participan millones de personas de todo el mundo. Cada grupo de discusión se centra en torno a un tema en particular, chistes, recetas de cocina, matemáticas, filosofía, computadoras, biología, ciencia ficción prácticamente cualquier tema que se pueda pensar tiene su propio grupo de discusión.

En total, *Usenet* tiene más de 5,000 grupos de discusión diferentes. Muchos de estos son de interés regional o local. Por ejemplo, existe un grupo de discusión para tratar sobre los restaurantes de la Bahía de San Francisco. Sin embargo, más de 2,500 grupos son de interés general y son leídos por personas de todo el mundo.

Cualquier tema de interés para la humanidad tiene cabida en algún grupo de discusión de *Usenet*. Cuando surge la necesidad de un grupo nuevo, hay unos procedimientos establecidos para formar uno. Al contrario que los servicios comerciales, no existe una autoridad central que controle *Usenet*. Por esta razón, cuando los usuarios deciden que es necesario crear un grupo, lo crean.

Este sistema tiene dos resultados importantes: Primero, los grupos nuevos pueden crearse justo cuando surge la necesidad. Segundo, hay una gran cantidad de grupos dedicados a temas esotéricos.

La idea original de la red *Usenet* fue concebida para distribuir anuncios y noticias. La idea fué crear una versión computarizada de un tablón de anuncios. *Usenet* pronto sobrepasó su idea original, pero el legado de una red de noticias electrónicas aún permanece. Aunque *Usenet* se utiliza principalmente para grupos de discusión, nosotros aún hablaremos utilizando términos de un servicio de noticias.



Por ejemplo, se hace referencia a *Usenet* como las *noticias*, o *noticias de la red*, aunque hay muy pocas noticias reales en el sentido de un periódico. De igual modo, los grupos de discusión de *Usenet* se denominan *grupos de noticias* o simplemente grupos. Dentro de cada grupo, cada una de las contribuciones individuales se llaman *artículos* o *anuncios*. Cuando se escribe un artículo para incluir en un grupo de discusión, se dice que se ha *enviado* un artículo.

Para leer artículos de *Usenet* se utiliza un programa denominado *lector de noticias*. El lector de noticias actúa como interfaz, le indicamos qué grupos de noticias queremos leer y nos muestra los artículos de ese grupo. Podemos leer cualquier grupo de noticias o de discusión, siempre y cuando se reciba en nuestra instalación local. Muchas instalaciones no disponen de todos los grupos de noticias, debido a que precisan mucho espacio en disco.

Existen en el mercado varios lectores de noticias y cada uno funciona de una forma propia. En los sistemas Unix, cuatro de los lectores de noticias más populares son *nn*, *rn*, *tin* y *trn*. Si se utiliza un sistema Unix, al menos debe estar disponible uno de estos programas. Una de las funciones principales de un programa de lectura de noticias es mantener la lista de grupos de discusión que se quieren leer. En cualquier caso, siempre se podrán añadir o eliminar grupos de la lista. Cuando se añade un grupo a la lista, se dice que nos hemos *subscrito* a ese grupo. Del mismo modo, cuando indicamos que no queremos seguir recibiendo las noticias de un grupo, se dice que *eliminamos la suscripción*.

Cuando eliminamos la suscripción o nos subscribimos a grupos y cuando leemos artículos, el programa de lectura de noticias mantiene un archivo donde almacena estos datos. Esto permite al lector de noticias conocer a qué grupos se está suscrito y saber qué artículos se han leído. En los sistemas Unix, este archivo se llama *.newsrsc* y se almacena en el directorio de inicio.

Mientras se está leyendo un artículo, el lector de noticias permite ejecutar diferentes acciones. Por ejemplo, se puede cambiar de un artículo a otro, grabar un artículo en un archivo, enviar por correo una respuesta a quien envió el artículo, redactar un artículo, etc. Cuando se leen las noticias, también se utiliza una estructura cliente/servidor. Los artículos son almacenados y administrados por un *servidor de noticias*. El lector de noticias actúa como el programa cliente. Cada vez que se escribe una orden para leer un artículo, el lector de noticias pide ese artículo al servidor de noticias.



En algunas redes muy grandes, como en las universidades, muchos usuarios desean leer las noticias a la vez. Si el servidor de noticias está muy ocupado, se pueden experimentar algunos retardos, mientras el lector de noticias espera para recibir un artículo.

*Usenet* se creó en 1979 en Carolina del Norte (E.U.) como un experimento. La idea fue crear un tablón de anuncios electrónico para facilitar el envío y lectura de los mensajes y noticias. Al comienzo, había solamente dos nodos: la Universidad de Carolina del Norte y la Universidad de Duke. Desde entonces, otros nodos se fueron adhiriendo, y con la explosión de las redes en los años 80's y los 90's *Usenet* se extendió rápidamente. Hoy en día, hay decenas de miles de nodos de *Usenet* en los que participan más de dos millones y medio de usuarios.

Cada nodo de *Usenet* es administrado por el *administrador de noticias*. En algunos lugares, el administrador de noticias es la misma persona que el administrador del sistema. Sin embargo, necesariamente éste no tiene que ser el caso. En las instalaciones muy grandes, como pueden ser las universidades, el administrador de noticias puede ser un miembro del personal informático o un usuario avanzado, que depende del administrador de sistemas.

Veamos un ejemplo típico en el que se leen artículos de uno de los grupos de discusión de *Usenet* dedicados a matemáticas. Mientras estamos leyendo el grupo de discusión, decidimos enviar un artículo. Veremos algunos detalles de como hacerlo y como se propaga el artículo a todo el mundo. Los detalles pueden variar de un sistema a otro, pero la idea general será la misma.

Para empezar, es necesario conectar con un host de **INTERNET** y ejecutar el programa de lectura de noticias. Se comienza diciéndole al lector de noticias qué grupo de discusión se quiere leer. El programa de lectura de noticias almacena en un archivo los grupos de discusión a los que se está suscrito y qué artículos se han leído.

En este caso, especificaremos el grupo de discusión de matemáticas, por lo que el lector de noticias comprueba los artículos que ya se han leído y luego conecta con el servidor de noticias para solicitar el artículo siguiente de este grupo en particular. El servidor de noticias responde enviando ese artículo, que será presentado en la pantalla por el lector de noticias.





Cuando se ha terminado de leer un artículo, el lector de noticias pide que se le envíe el siguiente y lo muestra en pantalla.

Como así puede ocurrir, algún usuario del otro lado del mundo ha enviado un artículo preguntando si alguien tiene una solución más corta para resolver el último teorema de Fermat. Precisamente el otro día obtuvimos una solución a ese problema. Bien, le indicaremos al lector de noticias que queremos redactar un artículo para responder al que estamos leyendo (esto se denomina un artículo "continuado"). Nuestro lector de noticias ejecuta el editor de textos y redactamos el artículo. Decimos que existe una solución muy elegante y corta a ese problema, pero que desafortunadamente, es demasiado larga para enviarla en el artículo.

Después de terminar de escribir el artículo, damos la orden de salir del editor. El lector de noticias recupera de nuevo el control y envía una copia del artículo al servidor central de noticias. En este punto, nuestro artículo puede ser leído por cualquier usuario local de nuestra red, sin embargo, el artículo todavía necesita ser enviado al resto del mundo.

De ahora en adelante, cuando un servidor de noticias suministra artículos de *Usenet* a otro servidor, diremos que *alimenta noticias*. De este modo para participar en *Usenet*, nuestro servidor de noticias obtiene artículos de otro nodo de *Usenet*. A determinados intervalos, nuestro servidor de noticias conectará con el servidor de ese nodo para obtener los artículos nuevos.

Cada vez que esto ocurre, nuestro servidor pasará cualquier artículo nuevo que aún no se haya recibido en el otro nodo. En particular, el artículo nuevo que hemos redactado se enviará fuera, más tarde, cuando el servidor de noticias conecte con su proveedor de noticias, nuestro artículo se enviará a otro nodo. Esta es la forma en que funciona *Usenet*: nuestro artículo se pasa automáticamente de servidor a servidor de noticias, cada vez que se haga una conexión.

Lo que hace que el sistema funcione tan rápidamente es que algunos servidores de noticias actúan como estaciones de paso, proporcionando artículos para muchos otros servidores. Una vez que nuestro artículo llegue a una de estas estaciones de paso, nuestro artículo será enviado a muchos servidores de noticias en un corto espacio de tiempo. Al día siguiente, el artículo estará disponible en casi todos los servidores de noticias del país.



En los dos o tres días siguientes estará disponible en todo el mundo, y al final de la semana, nuestro descubrimiento matemático será conocido en todo el mundo.

Ahora si enviamos una gran cantidad de artículos, al igual que los demás usuarios de *Usenet* podría darse el caso que se amontonaran indefinidamente, la solución es mantener cada artículo por un tiempo determinado y luego anularlo, cada administrador de noticias decide cuanto tiempo deberían mantenerse los artículos. El servidor de noticias comprueba regularmente los artículos y elimina aquellos cuya fecha es anterior al intervalo de tiempo especificado. Cuando sucede esto, se dice que los artículos han *expirado*.

De acuerdo con estadísticas propias de la red hay 76,000 nodos que reciben *Usenet*, y no todos pertenecen a **INTERNET**. En estos nodos, hay 8,445,000 usuarios de todo tipo, de los cuales alrededor del 29 por ciento (2,417,000) participan en *Usenet*, y en promedio, se transmiten 26,400 mensajes nuevos cada día, conteniendo un total de 56.2 MB (56.2 millones de caracteres) de datos.

## 📁 JERARQUIAS

Dentro de *Usenet*, existen diferentes jerarquías con el objeto de clasificar los diferentes tipos de artículos que se manejan dentro de ella.

Una primera clasificación o división en jerarquías dentro de *Usenet*, es la denominada jerarquía *principal* y *alternativa*, la tabla 3-1 muestra las jerarquías más importantes, y éstas se distribuyen por todo el mundo, a cada grupo de discusión se le asigna un nombre que consta de dos o más partes, separadas por puntos, la primera parte del nombre es la jerarquía a la que pertenece el grupo.

Las jerarquías principales se envían a todos los servidores de noticias de *Usenet*, mientras que las jerarquías alternativas se consideran opcionales. Muchos nodos de *Usenet* las reciben, pero otros deciden no hacerlo.



**Tabla 3-1 JERARQUIAS MAS IMPORTANTES DE LOS GRUPOS DE USENET.**

Nombre	Tema
alt	grupos alternativos, muchos temas diferentes
bionet	biología
bit	muchos temas: de las listas de correo de Bitnet
biz	negocios, mercadotecnia, anuncios comerciales
comp	computadoras
ddn	red de datos del Departamento de Defensa
gnu	Free Software Foundation y su proyecto GNU
ieee	Instituto de Ingenieros Eléctricos y Electrónicos
info	muchos temas: de las listas de correo de la Universidad de Illinois
kl2	de la guardería a la escuela secundaria
raisc	Cualquier tema que no se ajusta a ninguna categoría
news	grupos sobre <i>Usenet</i>
rec	recreo, pasatiempos, arte
sci	ciencias de todos los tipos
soc	publicaciones sociales
talk	debates sobre temas polémicas
u3b	computadoras AT&T 3B
vmsnet	computadoras DEC VAX/VMS y redes DECNET

En cada nodo de *Usenet*, el administrador de noticias decide qué jerarquías y qué grupos quiere recibir. En la mayoría de los nodos, cabe esperar ver todas las jerarquías principales y al menos algunas de las jerarquías alternativas. Sin embargo, se debe tener presente que si se recibe una jerarquía en particular, ésta puede que no contenga todos los grupos de esa jerarquía.

La principal diferencia entre los dos tipos de jerarquías depende de la forma en que se crean los grupos, los usuarios deben seguir procedimientos bien definidos para crear grupos principales. Debe haber una discusión y una votación, y muchos usuarios deben expresar su interés. Las jerarquías alternativas son más flexibles: cualquier usuario que sepa como crear un grupo de discusión puede hacerlo. Históricamente, las jerarquías principales fueron las categorías originales de *Usenet* y eran controladas de una forma muy estricta. Las jerarquías alternativas se crearon posteriormente y con menos restricciones. Como resultado, los grupos principales son más estables y bien aceptados por los administradores de noticias.



De todas formas, muchos de los grupos alternativos son muy populares y ampliamente aceptados. Sin embargo, como se puede imaginar, existen grupos de discusión bastante ridículos, especialmente en la jerarquía *alt*.

Los otros tipos de jerarquías que son: *culturales*, *organizativas* y *regionales*. Las jerarquías culturales contienen grupos dedicados a usuarios particulares. Por ejemplo, cuatro de las más conocidas son *de* (grupos de discusión alemanes), *rj* (japoneses), *aus* (australianos) y *relcom* (rusos).

Aunque la mayoría de los artículos se escriben en inglés, las jerarquías culturales a menudo contienen artículos en su propia lengua. Para leer estos artículos a veces será necesario un software especial para ver los caracteres no-ingleses.

Las jerarquías de organizaciones contienen grupos dedicados a una universidad, empresa u otra organización. Existen muchas jerarquías de este tipo, por ejemplo, la Universidad de California en Santa Bárbara tiene su propia jerarquía denominada *ucsb*. El Trinity College de Dublin, Irlanda tiene la jerarquía *tcd*.

Junto con las jerarquías culturales y de organizaciones, existe un gran número de jerarquías regionales. Estas contienen grupos de interés para las personas que viven en un zona determinada, por ejemplo, la jerarquía *ba* incluye muchos grupos pertenecientes al área de la Bahía de San Francisco, como *ba.markethousing*.

Muchas de estas jerarquías culturales, de organizaciones y regionales se distribuyen por todo el mundo. Por ejemplo, muchos nodos de *Usenet* reciben la jerarquía *fj*.

Dentro de *Usenet* existe un servicio real de noticias, se denomina Clarinet, y tiene su propia jerarquía, *clari*. El servicio Clarinet se suministra por una compañía privada. Al contrario que *Usenet*, si una organización quiere recibir Clarinet, debe pagar por ello. Sin embargo, el costo es bastante razonable y muchas organizaciones ofrecen este servicio a sus usuarios.



La jerarquía *clari* incluye muchos grupos de noticias diferentes, más de 270, algunos son de interés general, otros de carácter regional o local. La única diferencia que existe entre *Usenet* y *Clarinet*, es que los grupos de *Clarinet* son grupos de lectura únicamente; no es posible enviar artículos a estos grupos.

Si nuestro servidor de noticias no recibe la jerarquía *clari*, existe un grupo de *Clarinet* que es posible leer. Se llama *biz.clarinet.sample*. *Clarinet* envía regularmente artículos de interés a este grupo como ejemplo de su servicio.

Los nombres de los grupos de discusión son fáciles de entender. La primera parte del nombre indica la jerarquía, por ejemplo, un grupo que comience con *comp*, como *comp.unix.questions*, está en la jerarquía de computadoras, los grupos cuyos nombres comienzan con *talk* como *talk.environment*, son grupos de debate. En la jerarquía *alt* es posible encontrar grupos muy raros. Por ejemplo, existe un grupo denominado *alt.sex.stories*.

En consecuencia, cuando leemos una lista de grupos de noticias debemos de decidir cuales leer, comenzando por mirar la primera parte del nombre, que corresponde al de la jerarquía, siguen nombres de categorías y posiblemente subcategorías. Cuando dos grupos de noticias están relacionados, tienen nombres similares, diferenciándose únicamente en la última parte del nombre que es una letra *d*, significa que el grupo con la *d* sirve para discutir el contenido de otro grupo.

Otro tipo de grupos de noticias son los llamados *moderados*, lo que significa que no se pueden enviar artículos directamente a esos grupos. Los artículos se envían en la forma habitual, pero son automáticamente desviados y enviados a una persona, llamada *moderador*, que decide que artículos se incluyen en el grupo. Un moderador no sólo decide que artículos incluir en el grupo, a menudo edita y organiza los artículos. La función fundamental de los moderadores es minimizar el número de artículos de baja calidad en un grupo de noticias.

Algunos grupos de discusión moderados ofrecen otro tipo de envío especial denominado *recopilación*. El moderador crea una recopilación seleccionando todas las preguntas, respuestas y envíos sobre un tema determinado. El moderador edita esta información en una serie de temas de interés y lo envía en un gran artículo, la recopilación. Generalmente la terminación del nombre de estos artículos es *digest*.



Una recopilación es algo muy parecido a un número de una revista electrónica, con un volumen, número de ejemplar y una tabla de contenido. La mayoría de los programas de lectura de noticias tiene una orden especial que permite saltar de un tema a otro mientras se lee una de estas recopilaciones.

Algo curioso que ocurre dentro de **INTERNET** y de *Usenet* es el uso de acrónimos, algunos de los más comunes se enlistan a continuación.

### ACRONIMOS INTERNET

FAQ	preguntas más frecuentes
FAQL	lista de preguntas más frecuentes
FOAF	amigo de un amigo
MOTAS	persona del sexo apropiado
MOTOS	persona del sexo opuesto
MOTSS	persona del mismo sexo
MUD	dimensión múltiple
Ob-	[como prefijo] obligatorio
Objoke	chiste obligatorio
OS	sistema operativo
PD	dominio público
SO	otra persona importante (esposa, novio/a)
RTFM	lee el !".\$% manual

### ☞ LISTAS DE PREGUNTAS MAS FRECUENTES (FAQ)

Cuando se comienza a utilizar *Usenet*, surgen muchas preguntas, además, mientras se leen los distintos grupos de noticias surgirán nuevas preguntas sobre cada uno de esos temas. Muchas de estas preguntas serán las mismas que otros usuarios hayan formulado cuando empezaron a leer un grupo en particular, y aunque son interesantes para un principiante, los lectores experimentados se aburren de ver las mismas preguntas repetidamente.

La solución a este problema son las *listas de preguntas más frecuentes o FAQ*. Una lista FAQ es un documento, recopilado por un voluntario, que identifica y responde las preguntas que se realizan más a menudo en un grupo en particular. Muchos grupos de noticias tienen listas FAQ que son enviadas regularmente al grupo de noticias por la persona que las recopila.



Hay cuatro formas de obtener una lista FAQ. Primero, leer el grupo regularmente y si tiene una lista FAQ, se enviará cada cierto tiempo. Segundo, leer el grupo denominado *news.answers* que consta de listas FAQ y material relacionado. Este es un grupo muy interesante para leer, puesto que se verán listas de las mejores preguntas y respuestas sobre temas que normalmente no se leen. La tercera forma de obtener una lista FAQ es vía *ftp anónimo* para conseguir la lista desde un servidor de archivos de *Usenet*. Por último, después de probar los métodos anteriores, se puede enviar un artículo al grupo de noticias preguntando si existe una lista FAQ y de ser así, que nos envíen una copia por correo.

## ☞ INTERPRETACION DE LOS ARTICULOS DE *Usenet*

Un artículo de noticias consta de tres partes: una *cabecera*, seguida de un *cuerpo*, y finalmente una *signatura* opcional.

La cabecera contiene información técnica sobre el artículo, podemos encontrar hasta veinte líneas diferentes en una cabecera, cada una de las cuales contienen un tipo diferente de información. El cuerpo del artículo es el texto normal, es decir la parte principal del artículo. Y por último, la signatura consta de unas pocas líneas que se sitúan al final del artículo, las cuales se redactan por la persona que envió el artículo y se añaden automáticamente al final de cada artículo que se envía.

## ☞ LA CABECERA

Todos los artículos de *Usenet* deben tener una cabecera que conste de información técnica al principio del artículo. Cabe señalar que las líneas de cabecera serán o no visibles para los usuarios dependiendo del programa de lectura de noticias que se utilice y de como esté configurado.

Algunos lectores de noticias presentan por defecto todas las líneas de la cabecera. Otros, por el contrario, no lo harán a menos que se le indique de forma expresa; sólo mostrarán el cuerpo y la signatura del artículo. De todas formas, todos los lectores de noticias permiten configurar la forma en que se verán los artículos. Además, mientras se está leyendo un artículo, es posible pedir al lector de noticias que presente el artículo actual junto con su cabecera completa.



A continuación se describirá brevemente el significado de estas líneas.

- ◇ **Approved:** En párrafos anteriores, se explicó que algunos grupos de noticias son moderados. dentro de los cuales la línea de cabecera **Approved:** contiene la dirección de correo del moderador.
- ◇ **Control:** Esta línea de cabecera contiene órdenes especiales que utilizan los administradores de noticias para controlar *Usenet* y generalmente no aparece en los artículos normales, por ejemplo, existe un tipo de línea de control para crear un grupo nuevo.
- ◇ **Date:** Esta línea contiene la hora y la fecha de cuando se envió el artículo, la hora a menudo se indica respecto al Meridiano de Greenwich (GMT), que es el formato estándar en **INTERNET**. (Algunas veces GMT se denomina Hora Universal).
- ◇ **Distribution:** Cuando se envía un artículo, es posible especificar a donde se quiere enviar. Algunos programas de noticias presentarán varias opciones y pedirán que se seleccione una. Estas opciones pueden ser: la organización, la región, el país o el mundo entero. En algunos sistemas, no se verán estas opciones y debemos conocer que opciones hay disponibles. Un punto importante que hay que aclarar es que una línea **Distribution** no garantiza a donde se enviará el artículo, sólo es una sugerencia.
- ◇ **Expires:** Como ya mencionamos, cada administrador de noticias define el intervalo de tiempo que el servidor de noticias almacenará los artículos. La cabecera **Expires** se utiliza cuando se quiere recomendar una fecha de expiración distinta.





- ◇ **Followup-To:** Cuando queremos enviar un artículo en respuesta a otro artículo que hemos leído (lo que mencionamos como artículo continuado), normalmente lo hacemos al grupo o grupos en los que aparecía el artículo original, sin embargo en ciertos casos un usuario puede querer tener el control sobre donde deberían ir los artículos continuados. Para estos casos, la línea **Followup-To** especifica a qué grupos deberían enviarse estos artículos.
  
- ◇ **From:** Esta línea es muy importante, ya que indica quién envió el artículo. Siempre aparecerá la dirección del usuario y en muchos casos también aparecerá su nombre completo. Quizá lo más importante del contenido de esta línea es que existen algunas computadoras que toman los datos de ésta y los utilizan para los servicios de páginas blancas para *Usenet*.
  
- ◇ **keywords:** Esta línea contiene una o más palabras o términos que categorizan el contenido del artículo, algunos usuarios comprueban el contenido de esta línea para decidir si leen el artículo.
  
- ◇ **Lines:** Esta línea de cabecera revela el número total de líneas que tiene el artículo, que incluye el cuerpo y la signatura, pero no la cabecera.
  
- ◇ **Message-ID:** El contenido de esta línea es un identificador único que se genera automáticamente por el programa que envía el artículo, la última parte del número es la dirección de la computadora que envió el artículo. Esta información sólo se utiliza en los programas más modernos.
  
- ◇ **Newsgroups:** Esta línea muestra a qué grupos se enviará el artículo, cuando un artículo se envía a más de un grupo, éstos aparecerán en la línea de cabecera **Newsgroups**.



- ◇ **NNTP-Posting Host:** El sub-protocolo de TCP/IP que se utiliza para transportar los artículos de *Usenet* se denomina NNTP (Network News Transfer Protocol, Protocolo de Transferencia de Noticias de Red), la línea de cabecera NNTP-Posting Host indica el nombre de la computadora desde la que se envió el artículo, es decir que indica que computadora utilizó la persona que envió el artículo.
- ◇ **Organization:** Esta línea de cabecera contiene una breve descripción de la organización a la que pertenece el usuario que envió el artículo o la organización a la que pertenece la computadora.
- ◇ **Path:** La información de esta línea consiste en un número de nombres de computadoras, separados por el carácter ! (exclamación). Los nombres muestran el camino que siguió el artículo, de computadora en computadora, hasta llegar a nuestro servidor de noticias, para lo cual debemos leerlo de derecha a izquierda.
- ◇ **References:** Esta línea solamente aparecerá en los artículos continuados. Si aparece, contendrá el identificador de la línea Message-ID del artículo original. Además se utiliza por el programa de lectura de noticias para agrupar los artículos continuados. De este modo, después de leer un artículo, el lector de noticias puede mostrar un artículo continuado tras otro. A la serie de artículos continuados se le denomina hilo.
- ◇ **Reply-To:** Esta línea de cabecera tiene el mismo formato que la línea From, si existe indica que las respuestas por correo deben remitirse al usuario que envió el artículo en la dirección que aparece en esta línea.
- ◇ **Sender:** El propósito de esta línea es indicar el nombre de la computadora desde la que se envió el artículo y su contenido se genera de forma automática independientemente del contenido de la línea From.



- ◇ **Subject:** Esta línea contiene una breve descripción sobre el tema del artículo. Esta descripción se crea cuando se redacta el artículo. Algunos programas de lectura de noticias permiten a los usuarios seleccionar que artículos quieren leer basándose en el contenido de la línea Subject.
- ◇ **Summary:** Esta cabecera proporciona una breve descripción resumiendo el artículo, lo que es muy útil en los artículos continuados.
- ◇ **Xref:** Cuando un artículo se envía a más de un grupo (envío cruzado), la línea Xref indicará los grupos de noticias a los que se envía el artículo, así mismo indica los números locales del artículo que lo identifican en cada grupo.

## ↳ LA SIGNATURA

La signatura es un aditamento opcional que muestra información sobre el usuario que envió el artículo. La mayoría de los usuarios tienen una misma signatura y la guardan en un archivo, y el programa de envío de noticias la añadirá automáticamente al final de cada artículo antes de enviarlo.

Para sistemas Unix, se debe de crear mediante un editor de textos un archivo llamado *.signature* en el directorio de inicio, y siempre que se envíe un artículo, el programa de noticias comprueba si existe este archivo. Si es así, el programa añade el contenido del archivo al final del artículo.

La información estándar para incluir en la signatura es el nombre de la persona que envió el artículo y su dirección de correo, opcionalmente se puede añadir el nombre de la organización a la que se pertenece, una dirección postal y quizás un número de teléfono. Después de esto, se puede colocar alguna frase de algún famoso escritor, nuestro lema de la vida, frases ingeniosas, dibujos, chistes, etc.



Article 123 (534 more) in alt.newbies  
From: fflintstone@cedrock.com  
Newsgroups: alt.newbies alt.newusers  
Subject: IS ANYONE OUT THERE?  
Date: 26 Jun 1994 13:45:12  
Lines: 8

HELLO. IS ANYONE OUT THERE READING THIS MESSAGE?

IM REALLY INTO MOUNTAIN BIKING. ITS SO COOL!!!!

FRED

Article 145 (345 more) in alt.newbies  
From: rmaclean@bnr.ca.edu  
Newsgroups: alt.newbies  
Subject: Re: IS ANYONE OUT THERE?  
Date: 26 Jun 1994 20:12:34  
Lines: 12

> HELLO. IS ANYONE OUT THERE READING THIS MESSAGE?

Only a few million of us! (Well, those of us who subscribe to this newsgroup, at any rate.)

> IM REALLY INTO MOUNTAIN BIKING. ITS SO COOL!!!!

Since you are obviously a newbie, you won't get flamed too much. Try checking out newsgroups under rec.bicycles, such as rec.bicycles.tech, rec.bicycles.misc, or rec.bicycles.soc (there are others).

> FRED

Nice to meet you Fred. By the way, there's no need to shout. Check out the posting guidelines in new.announce.newusers.

I'd rather  
be sailing

Roy MacLean  
rmaclean@bnr.ca.edu



## EL LECTOR DE NOTICIAS COMO UN CLIENTE

Como la mayoría de los servicios de **INTERNET**, las noticias de *Usenet* utilizan la estructura cliente/servidor. En este caso, el servidor de noticias reside en una de las computadoras de la red, su trabajo es mantener el flujo de noticias hacia el mundo exterior y gestionar un gran número de artículos y grupos de noticias que se reciben diariamente.

Para leer las noticias, se utiliza un programa cliente, denominado *lector de noticias*. El lector de noticias actúa como interfaz entre el usuario y *Usenet*, permitiendo seleccionar los grupos de noticias a los que se desea subscribir, elegir los artículos que se quieren leer de esos grupos, y mostrarlos por la pantalla. Mientras se leen los artículos, se pueden almacenar en un archivo, enviarlos por correo a otro usuario o imprimirlos. También se puede responder al artículo, enviando un mensaje privado a su autor, o enviando un artículo público al grupo de noticias.

Ahora bien, las noticias se transportan sobre **INTERNET** utilizando el protocolo NNTP. Cuando los artículos llegan a un servidor de noticias, se almacenan en un formato estándar y para leer las noticias, todo lo que se necesita es un programa de lectura de noticias que conozca como tratar ese formato. Hay muchos lectores de noticias diferentes, diseñados para operar en una gran variedad de computadoras, sin embargo, la mayoría de los usuarios que utilizan a **INTERNET** manejan computadoras Unix, por lo que quizá los lectores de noticias más utilizados sean los que ofrecen los sistemas Unix, no obstante, el principio de funcionamiento es el mismo.

Para mantener los artículos de cada grupo de noticias, el servidor de noticias asigna un identificador único a cada artículo. El número se asigna por orden de llegada de los artículos. Los números empiezan por 1, siempre que se recibe un artículo nuevo, se le asigna un número secuencial, algunos lectores muestran este número cuando se leen los artículos, y otros en cambio no lo hacen y una vez que un artículo expira su número se elimina del sistema.



Cada grupo de noticias, posee su propia numeración, por lo que es posible que existan números de artículos iguales en grupos diferentes. El programa de lectura de noticias utiliza estos números para guardar la pista de los artículos que el usuario ha leído de cada grupo. Para ello, mantiene un archivo denominado *.newsrc* en el directorio de inicio de cada usuario, el cual contiene los nombres de cada grupo de noticias. Para cada nombre, esta información indica si se está suscrito al grupo y que artículos se han leído.

Como ya mencionamos, existe una gran variedad de programas que nos permiten leer las noticias, de los más populares tenemos *rn*, *trn*, *nn* y *tin* para terminales ASCII; *xrn* y *xvnews* para X-Window; y *gnus* y *gnews* para el entorno Emacs.

***rn*** es el más antiguo de los programas de lectura de noticias, fue creado por Larry Wall de System Development Corporation en Santa Mónica, California y la primera versión se creó el 8 de Abril de 1983. Hoy, *rn* es mantenido y desarrollado por Stan Barber, del Colegio de Medicina Baylor en Houston, Texas. Wall desarrolló *rn* para reemplazar otro lector de noticias llamado *newsread*. Diseñó cuidadosamente *rn* para reducir al máximo la interacción entre el programa y el usuario, de forma que la información se presentara de la forma más rápida posible. El resultado fué un lector de noticias que muestra tantos artículos como es posible en la mínima cantidad de tiempo.

***trn*** fué creado por Wayne Davison de Borland Corporation, en Scott Valley, California. *trn* es una variación moderna de *rn* y su primera versión data del 21 de julio de 1990, posee todas las características y órdenes de *rn* junto con algunas características adicionales, al contrario que *rn*, *trn* agrupa los artículos por temas (hilos) y hace mucho más fácil seleccionar que hilo se quiere leer. (un hilo es una serie de artículos que tratan del mismo asunto: esto es, el artículo original y todos los artículos continuados).



***nn*** fué desarrollado por Kim Storm de Texas Instruments A/S de Dinamarca, La primera versión salió en 1984 en Alemania, en 1988 en Europa y en 1989 para todo el mundo. Hoy en día, Peter Wemm de Demaas Popietary Limited de Australia es quien mantiene *nn*. Storm diseñó *nn* coincidiendo con el gran crecimiento que experimentó *Usenet* de forma que se recibían muchos más artículos de los que era posible leer, por esta razón se diseñó *nn* de forma que resultara fácil examinar un gran número de artículos muy rápidamente, en contraparte, *nn* requiere más interacción con el usuario que *rn* o *trn*.

***tin*** es el lector de noticias de más reciente creación de los cuatro, Fué creado por Lain Lea y su primera versión es del 23 de Agosto de 1991. En esa época, *Usenet* había crecido de forma exagerada. No solamente había muchos más artículos de los que era posible leer por un usuario, sino que había también demasiados grupos de noticias. Lea diseñó *tin* de forma que permitiera examinar la enorme lista de grupos de noticias existentes y seleccionar aquellos grupos que se quisieran seguir, una vez que se ha elegido un grupo, *tin* muestra la lista de artículos que componen ese grupo y permite seleccionar aquellos artículos que se desean leer.

En otras palabras, *rn* presenta los artículos de forma rápida con la mínima intervención del usuario, *trn* trabaja de forma análoga a *rn* sólo que permite seguir de forma fácil un hilo entero, *nn* requiere más intervención del usuario, pero permite examinar un gran número de artículos rápidamente y finalmente *tin* es el que más intervención del usuario requiere, pero permite manejar un gran número de grupos de noticias y, dentro de cada grupo examinar un gran número de artículos.

## ☐ CONVERSACIONES DENTRO DE INTERNET

### ☞ EL PROGRAMA TALK

El programa *talk* permite conectar nuestra computadora dentro de **INTERNET** con cualquier otra computadora e intercambiar mensajes. Cuando nos comunicamos de esta forma, se dice que estamos *conversando* con la otra persona, y lo que hacemos al conversar en **INTERNET** es ver lo que la otra persona escribe tan pronto como él o ella pulsan una tecla.



Lo importante del programa *talk* es que se puede conversar con gente de todo el mundo. Y cuando se conversa con alguien, la comunicación real la maneja un programa llamado *demonio talk*. Un demonio es un programa que se ejecuta en segundo plano, para proporcionar un servicio de interés general. En este caso, el demonio proporciona un servicio de conversación.

Para comenzar una conversación se escribe *talk* seguido de la dirección de la persona con la que deseamos conversar. El programa enviará un mensaje a la computadora de nuestro amigo y le dirá que queremos conversar con él, una vez que llegue la petición, deberá de contestarla y de no hacerlo nuestra máquina seguirá enviándole un mensaje cada diez segundos. Si decidimos dejarlo, simplemente hay que pulsar CTRL-C para parar el programa *talk*. La forma de contestar es introduciendo la orden *talk* seguida de la dirección desde la cual proviene la llamada.

Una vez que el demonio *talk* ha hecho la conexión con el demonio *talk* de otra persona, veremos un mensaje que nos indica que la conexión se ha establecido y al momento de acceder se trazará una línea horizontal en el medio de la pantalla, dividiéndola en mitad superior y mitad inferior. Todo lo que escribamos se verá en la mitad superior, todo lo que escriba la otra persona se verá en la mitad inferior.

Lo interesante de *talk* es que ambas personas pueden escribir al mismo tiempo y lo que se escribe se verá inmediatamente en ambas pantallas. Al escribir, se puede pulsar RETROCESO para corregir el caracter previo. Se puede usar también CTRL-W para borrar una palabra completa y CTRL-U para borrar una línea completa. En cualquier momento, se puede decir a *talk* que reescriba la pantalla completa pulsando CTRL-L. Esto puede ser cómodo si aparece en la pantalla un mensaje inesperado, y la conversación durará tanto como queramos.

Una de las consideraciones que debemos hacer al conversar a través de **INTERNET**, es que no podemos apreciar las expresiones corporales propias de la comunicación oral y que no podemos escribir tan rápido como hablamos, por lo que para compensar estas limitaciones utilizamos ciertas abreviaturas que se usan comúnmente para acelerar las cosas, algunas de las cuales se mencionan a continuación:





BCNU	estoy viéndote
BTW	por el momento
BYE	adiós (estoy listo para terminar de hablar)
BYE?	¿adiós? (¿estás listo para terminar de hablar?)
CU	adiós, hasta la vista
CUL	nos vemos luego
FYI	para tu información
FWIW	por lo que vale
GA	adelante y escribe (esperaré)
IMHO	en mi modesta opinión
IMO	en mi opinión
JAM	un minuto
O	cambio (tu turno para hablar)
OO	cambio y corto (adiós)
OBTW	oh, por el momento
ROTFL	muerto de risa
R U THERE?	¿estás ahí?
SEC...	espera un segundo
WRT	con respecto a

El segundo convenio importante reconoce que es fácil ofender a alguien (o al menos construir mal las frases) cuando no se está conversando en persona, en la computadora no somos capaces de usar un lenguaje sutil para indicar que lo que decimos es cortés e irónico, y no realmente tan tonto e insensible como parece.

El último consejo es que nadie se preocupa mucho por la ortografía, de hecho es molesto observar que nuestro interlocutor se pasa la mitad del tiempo corrigiendo las faltas de ortografía que se generan por escribir rápido en el teclado.

Dentro de las posibilidades de conversación, existe un problema que surge regularmente y es que no todas las computadoras de **INTERNET** pueden usar *talk* para conectarse unas con otras.



Puede suceder que nuestro programa *talk* sea incompatible con el de nuestro interlocutor, es decir que sus demonios son incompatibles. De forma breve, los datos de una computadora se almacenan en bytes, cada byte contiene un caracter y dentro de la computadora, los bytes se combinan en palabras. Puede haber dos bytes por palabra, cuatro bytes por palabra, o los que sean, lo importante es saber en que orden se almacenan los bytes dentro de cada palabra, por ejemplo si tenemos una palabra que contiene dos bytes, llamados A y B, algunas computadoras almacenan los bytes como AB mientras que otras computadoras los almacenan como BA. Los dos esquemas que hemos descrito se llaman *big-endian* y *little-endian*. Las computadoras Sun y Macintosh son *big-endian*. Las computadoras VAX y los PC son *little-endian*.

A esta incompatibilidad a veces se le llama problema *NUXI*, ya que la palabra "UNIX" en una computadora se almacenaría como "NUXI" en otra.

Para resolver este problema, se ha desarrollado otra versión de *talk*, llamada *ntalk*. El programa *ntalk* puede conversar con cualquier programa *ntalk*, independientemente de en que computadora esté corriendo. Muchos administradores de sistemas han instalado *ntalk* en sus máquinas, así que si tenemos problemas con *talk*, se puede intentar la misma orden de nuevo usando *ntalk*. Ahora bien, algunos administradores de sistemas instalan *ntalk* con ese nombre, de tal forma que cuando quieren la nueva versión, usan *ntalk* y cuando quieren la antigua usan *talk*. Otros administradores de sistemas instalan *ntalk* bajo el nombre *talk* y renombran la versión original como *otalk*. En estos sistemas se usa *talk* para obtener la nueva versión y *otalk* para obtener la versión antigua.

La sugerencia es entonces experimentar con *utalk*, *talk* y *otalk* hasta que encontremos la combinación que funciona.

#### ☞ EL PROGRAMA *ytalk*

La solución real a todo esto es usar un programa mucho mejor llamado *ytalk*, no sólo *ytalk* funcionará con cualquier otra computadora independientemente si ésta corre *talk*, *ntalk* o *ytalk* sino que además permite conversar a más de una persona a la vez.



El programa *ytalk* ha sido escrito por Britt Yenne como una sustitución al programa estándar de Unix *talk*. Lo bueno de *ytalk* es que funciona con ambas versiones de *talk* que hemos discutido antes en esta misma sección, si se usa *ytalk*, nunca hay que preocuparse sobre qué tipo de *talk* usan las demás personas con las que nos queremos comunicar, además si sabemos que otras personas utilizan *ytalk*, podemos usarlo para mantener una conversación en grupo. Cuando conversamos con una persona, la pantalla se divide en dos partes, como con *talk* pero con una diferencia importante *ytalk* mostrará el identificador de usuario de la persona con la que estamos conversando. Lo que escribimos se muestra en la mitad superior y lo que escribe la otra persona se muestra en la mitad inferior.

Si se quiere conversar con más de una persona, todo lo que hay que hacer es especificar más de una dirección como parte de la orden. Y en estos casos, *ytalk* dividirá la pantalla en tantas particiones como sea necesario. En cualquier momento, se pueden cambiar los participantes llamando a un menú especial con la tecla ESC.

Para tener una conversación en grupo, todo el mundo necesita estar usando *ytalk*, sin embargo, si se tiene *ytalk*, todavía se puede conversar con quien usa el programa *talk* normal. Si bien es cierto que él sólo podrá conversar con nuestra máquina y con nadie más.

Por ejemplo, digamos que tenemos *ytalk* y que queremos conversar con dos personas que sólo tienen *talk*. Si nos conectamos con ambos, veremos una conversación entre tres, pero cada uno de ellos, sólo verán la pantalla estándar de *talk* con una conversación entre dos personas. Ellos serán capaces de conversar con nosotros, pero no entre ellos.



-----= YTalk Version 2.0 =-----

Lo que escribamos se vera aquí.

Igual que la orden talk normal,  
uno puede escribir al mismo tiempo que  
la otra persona.

-----= rick =-----

Lo que la otra persona escribe se ve aquí.  
El puede escribir al mismo tiempo que nosotros y ambas  
partes de la pantalla se actualizaran  
simultáneamente

## ☰ INTERNET RELAY CHAT (IRC)

### ☞ ¿QUE ES IRC?

En la sección anterior discutimos dos formas de conversar directamente con otros usuarios de **INTERNET**. Se puede usar el programa *talk* y conversar con una persona o se puede usar *ytalk* y conversar con una o más personas a la vez. Lo que estos dos programas tienen en común es que las conversaciones son privadas tal y como sucede en una llamada de teléfono, ya que controlamos con quien conversamos y sólo esas personas pueden participar en la conversación. Una charla ofrece un tipo diferente de experiencia en la cual se conversa con mucha gente de toda la **INTERNET**.

Participar dentro de *IRC* es como estar en una fiesta, con un número elevado de personas y todas mezcladas entre si, las cuales tienen diversas conversaciones por lo que, todo lo que decimos lo puede oír toda la gente de nuestro grupo, y también podemos escuchar las conversaciones de otros. Si quisiéramos podemos ir caminando de un grupo a otro, unirnos a su conversación o simplemente escuchar, también podemos invitar a alguien (o a varios), a ir a un rincón y tener una conversación más privada. Y, si surge la necesidad, se puede susurrar un mensaje personal en el oído de alguien. Esto es **INTERNET Relay Chat** o *IRC*.



*IRC* fué desarrollado en 1988 por Jarkko Oikarinen en Finlandia. La versión actual de *IRC* se llama *IRC II*. Desde el principio, *IRC* se ha convertido en uno de los recursos más populares de **INTERNET** y es frecuentado por personas de muchos países diferentes.

## ☞ FUNCIONAMIENTO

Para operar *IRC* necesitamos ejecutar un programa cliente que actúe como interfaz, el cual se conecta con un servidor *IRC*, una vez habilitado nuestro cliente entonces podemos unirnos a un grupo de gente o movernos de un grupo a otro, hablando y escuchando según nos plazca.

Cada servidor *IRC* está conectado a los servidores más cercanos, de esta manera, todos los servidores *IRC* están conectados (al menos indirectamente) unos con otros. Así, cuando hacemos el contacto, estamos conectados a una telaraña de usuarios *IRC*, todos ellos hablando veinticuatro horas al día, encontraremos un gran número de personas de muchos países diferentes, todos ellos conectados a *IRC*, hablando, hablando, hablando.

Para poner en orden lo que de otra manera sería un caos conversacional, *IRC* mantiene un número de diferentes *canales*. Cuando nos conectamos a *IRC* primero elegimos a qué canal nos incorporamos, y entonces ya podemos hablar con otras personas del mismo canal. Cuando queremos cambiar, podemos dejar el canal y unirnos a otro. Si queremos, podemos incorporarnos a más de un canal a la vez.

En cualquier momento, encontraremos varios miles de personas, participando en cientos de canales diferentes. Cuando nos movamos de un canal a otro, encontraremos gente de todo el mundo. La mayoría de las conversaciones son en Inglés, pero hay muchas en otros idiomas.

Para evitar que las cosas sean demasiado confusas, *IRC* usa dos reglas. La primera, cada canal tiene su propio nombre. La mayoría de los nombres de canales empiezan con un caracter #, algunos de los cuales son para discutir sobre temas específicos, otros en cambio son para hablar de lo que surja. Hay canales públicos, canales privados y canales secretos o "invisibles".



¿Cómo se puede crear un canal? Siempre que se da la orden para incorporarse a un canal, *IRC* comprueba si ese canal ya existe, de ser así, nos incorporamos a él y en caso contrario, *IRC* creará un nuevo canal usando el nombre que especificamos. Al principio, seremos los únicos en el canal, pero podemos esperar a que otras personas se unan a nosotros. Cuando la última persona abandona el canal, *IRC* lo elimina.

La primera persona que se incorpora a un canal (y por tanto, lo crea) obtiene la condición de *operador del canal*. El operador tiene control sobre varios aspectos del canal, por ejemplo, él o ella pueden hacer que el canal sea privado, de forma que sólo las personas que sean invitadas se puedan unir a él, también posee la capacidad de extender la condición de operador de ese canal a otra persona y poner el tema de discusión.

La otra convención es que cada participante en *IRC* tiene un apodo, cuando se empieza una sesión *IRC*, hay que especificar un apodo de hasta nueve caracteres, el cual se puede cambiar en cualquier momento. La única limitación es que no se puede elegir un apodo que se esté utilizando en ese momento. Si se quiere un apodo permanente, se puede registrar en una base de datos especial de *IRC*, de forma que siempre seamos conocidos por el mismo nombre.

En todo momento, el cliente *IRC* actúa como una ventana, mostrando lo que está diciendo todo el mundo en el canal. Con la ventaja de que lo que se escribe se muestra en la última línea, pero no se transmite realmente hasta que se pulsa RETURN, lo cual nos permite hacer correcciones mientras se escribe.

La penúltima línea es la línea de estado, que nos indica nuestro apodo, el nombre del canal, la hora, y un recordatorio de como ver información de ayuda. El resto de la pantalla se deja para la conversación real. Cuando miramos la pantalla, vemos lo que está escribiendo todo el mundo en nuestro canal. Irá apareciendo línea tras línea al final del área de visualización. Cuando aparecen nuevas líneas, las antiguas se desplazan hacia arriba. Así, una experiencia *IRC* es ver muchos mensajes, uno tras otro.

La mejor forma de aprender como usar *IRC* es leer algo sobre él, empezar por el *IRC primer* y el *IRC tutorial*, ambos disponibles por *ftp anónimo*. También debemos leer la lista de preguntas más frecuentes (*FAQ*) sobre *IRC*.



Ahora bien mencionamos que para usar *IRC*, hay que ejecutar un programa cliente de *IRC*. El cliente se conectará automáticamente a un servidor *IRC* y al resto del mundo. Si no tenemos un cliente *IRC*, podemos cargar un cliente *IRC* desde un servidor de *ftp anónimo*. La lista de *FAQ* nos revelará donde buscarlo. Hay varios clientes disponibles para diferentes tipos de computadoras, y una vez en nuestra computadora deberemos instalarlo.

También hay clientes *IRC* de acceso público que se pueden usar vía *telnet*, hay que buscar en la lista de *FAQ* sobre *IRC* una dirección y hacer *telnet* a esa dirección, seguramente a través de un puerto virtual (Esto lo veremos más adelante), una vez conectados se ejecutará automáticamente un programa cliente *IRC*.

Una vez conectados a *IRC*, hay un gran número de órdenes que se pueden usar, las cuales comienzan con el símbolo */* (barra inclinada) y debe ser el primero de la línea. Cuando se introduce una orden, aparecerá en la última línea de la pantalla y hasta que presionemos RETURN, la orden se envía a *IRC*. Siempre que se escriba una línea que no empiece con la diagonal, será considerada parte de una conversación (una línea normal) y se enviará al canal.

Como referencia, la Tabla 3-3 contiene un sumario de las órdenes básicas de *IRC*.



**Tabla 3-3.** Resumen de las órdenes básicas de *IRC*

Orden	Descripción
/flush	descargar la salida restante de una orden
/help	mostrar una lista de órdenes de <i>IRC</i>
/help <i>orden</i>	mostrar ayuda sobre una orden específica
/help intro	mostrar una introducción sobre <i>IRC</i>
/help newuser	mostrar información para usuarios nuevos
/join <i>canal</i>	unirse al canal especificado
/leave <i>canal</i>	abandonar el canal especificado
/list	mostrar información de todos los canales
/list <i>canal</i>	información del canal especificado
/list -min <i>n</i>	mostrar los canales con al menos <i>n</i> personas
/list -max <i>n</i>	mostrar los canales con no más de <i>n</i> personas
/me <i>accion</i>	mostrar la acción especificada
/mode * +pi	hacer privado el canal actual
/msg <i>apodos texto</i>	enviar un mensaje privado a la persona especificada
/msg , <i>texto</i>	enviar un mensaje a la última persona que nos envió un mensaje
/msg . <i>texto</i>	enviamos un mensaje
/nick	mostrar nuestro apodo actual
/nick <i>apodo</i>	cambiar nuestro apodo al nombre especificado
/query <i>apodos</i>	enviar todos nuestros mensajes a las personas especificadas
/query	terminar de enviar mensajes privados
/quit	salir de <i>IRC</i>
/set novice off	permitir ciertas acciones (como unirse a múltiples canales)
/who <i>canal</i>	mostrar quien está en el canal especificado
/who <i>apodo</i>	mostrar información sobre la persona especificada
/who *	mostrar quien está en el canal actual
/whois <i>apodo</i>	mostrar toda la información sobre la persona especificada
/whois *	mostrar toda la información sobre todo el mundo





# INTERNET

---



## ***Usenet* : Grupos de Noticias**

Son grupos de discusión, en torno a un tema en particular,  
los temas pueden ser de interés global, regional o local.

Cliente: Lector de noticias  
rn, trn, nn, tin para terminales ASCII  
xm, xvnews para X-Window

Notas:

# INTERNET



## **Usenet Terminología**

- ⌘ Noticias de la Red.
- ⌘ Artículos o Anuncios.
- ⌘ Lector de Noticias.
- ⌘ Suscripción (y su eliminación).
- ⌘ Servidor de Noticias.
- ⌘ Administración de noticias.
- ⌘ Artículo: Continuado/expirado.
- ⌘ Moderador.

**Notas:**

# INTERNET



## **Usenet: Jerarquías**

Organizan los grupos de discusión.

<u>Nombre</u>	<u>Tema</u>
alt	Grupos alternativos, muchos temas diferentes.
bionet	Biología.
biz	Negocio, mercadotecnia, anuncios comerciales.
comp	Computadoras.
ddn	Red de datos del Departamento de Defensa
ieee	Instituto de ingenieros Eléctricos y Electrónicos.
kl2	De la guardería a la escuela secundaria.
raisc	Cualquier tema que no se ajusta a ninguna categoría.
news	Grupos sobre <i>Usnet</i> .
sci	Ciencia de diversas áreas.
soc	Publicaciones sociales.
talk	Debates sobre temas polémicos.
u3b	Computadoras AT&T 3B.

**Notas:**

# INTERNET

---



## **Usenet: Ejemplos de Nombres:**

↳ comp.unix.questions.

↳ bionet.ADN

↳ comp.software.was

↳ alt.sex.stories

existen listas FAQ.- Lista de preguntas más frecuentes.

**Notas:**

# INTERNET

---



## ***Usenet:* Estructura del Artículo**

↵ Cabecera

↵ Cuerpo

↵ Signatura

**Notas:**



## **Conversaciones dentro de Internet**

### **TALK**

↳ Conversación interactiva entre dos usuarios.

↳ Variantes: big-endian y little-endian.

( UNIX )      ( NUXI )

### **YTALK**

↳ Conversación interactiva entre varios usuarios.

**Notas:**

# INTERNET



## Conversaciones dentro de Internet

----- YTalk Version 2.0 -----  
Lo que escribamos se verá aquí.

Igual que la orden talk normal,  
uno puede escribir al mismo tiempo que  
la otra persona.

----- rick -----  
Lo que la otra persona escribe se ve aquí.  
El puede escribir al mismo tiempo que nosotros y ambas  
partes de la pantalla se actualizarán  
simultáneamente

Notas:

# INTERNET

---



## **Conversaciones dentro de Internet.**

**IRC**

*Internet Realy Chat.*

**Se participa dentro de una "fiesta", donde existen diversas conversaciones, se pueden establecer canales privados.**

**Notas:**



# CURSO: **INTERNET**

## 4.- TELNET Y ftp



Octubre de 1995.

## 4.- *telnet* Y *ftp*

### ☐ *telnet*: SESION REMOTA INTERACTIVA

#### ☞ INTRODUCCION

Lo más asombroso de **INTERNET**, es la facilidad para utilizar una computadora al otro lado del mundo como si fuera la del otro lado del pasillo, vamos a discutir a lo largo de esta sección como establecer sesiones remotas.

Se tratarán las dos formas principales de usar una computadora remota. Primero, se puede conectar con una computadora en la que se tenga una cuenta, por ejemplo para una computadora Unix, significa que debemos tener un identificador de usuario y una palabra clave, y una vez iniciada la sesión, se puede utilizar la computadora normalmente.

Segundo, hay muchas computadoras de **INTERNET** que ofrecen algún tipo de servicio a través de *telnet*, dichos sistemas están disponibles para cualquier usuario y normalmente no requieren una palabra clave.

Como podrá verse, *telnet* es tan transparente que resulta fácil olvidar que se está lejos de la computadora remota. La principal limitación es que, cuando el tráfico de la red **INTERNET** es muy alto, la respuesta de una computadora muy alejada puede retardarse significativamente, a pesar de eso, es común que los usuarios experimentados trabajen con varias computadoras diferentes de **INTERNET**, moviéndose de una a otra fácilmente.

Para utilizar el servicio de sesion remota debemos ejecutar un programa especial llamado *telnet* en nuestra computadora. Este programa utiliza **INTERNET** para conectarse a la computadora especificada. Una vez que se hace la conexión, *telnet* actúa como un intermediario entre nosotros y la computadora remota. Cada vez que se pulsa una tecla en nuestra computadora, ésta se envía a la computadora remota. Cada vez que la computadora remota produce una respuesta, se envía a nuestra computadora que la muestra en la pantalla. El resultado final es que nuestro teclado y nuestra pantalla parecen estar conectados directamente a la otra computadora.



En la terminología *telnet*, nuestra computadora se denomina la computadora local, mientras que la otra computadora (con la que el programa *telnet* nos conecta) se denomina computadora remota. Se utilizan estos términos sin importar la distancia a que se encuentren las computadoras, tanto si están al otro lado del mundo o en la misma habitación.

Como se mencionó anteriormente, a menudo se hace referencia a las computadoras de **INTERNET** como host. Pero dentro de la terminología *telnet*, se puede decir que el trabajo del programa es conectar nuestra computadora local a un host remoto de **INTERNET**.

#### ✓ EL PROGRAMA *telnet*

Existen dos formas de ejecutar *telnet*, una de ellas es escribir el comando seguido de la dirección a dónde nos queremos conectar y la otra es escribir solamente el comando y ejecutar posteriormente algunas instrucciones, vamos a comentar a lo largo de la sección ambas formas.

Sabemos que todos los hosts de **INTERNET** tienen una dirección oficial conocida como dirección IP, la cual consta de varios números separados por puntos. No obstante la mayoría de las veces ejecutamos el comando *telnet* seguido del nombre del host al que nos queremos conectar en lugar de utilizar su dirección IP, pero podemos dar también la instrucción con ésta dirección. Algunos sistemas tienen problemas utilizando determinadas direcciones estándar, si se encuentra este problema con *telnet*, debe probarse con la dirección IP.

Cuando el programa *telnet* se ejecuta, iniciará una conexión con el host remoto especificado. Mientras *telnet* está esperando una respuesta se verá en pantalla un mensaje que nos indica que *telnet* está intentando comunicarse, y una vez que se establece la conexión que puede llevar unos segundos si el host está muy distante aparecerá un mensaje que nos indica el lugar al que nos hemos conectado.



Si, por alguna razón, *telnet* no puede realizar la conexión se mostrará un mensaje diciendo que el host es desconocido, pero existen varias razones por las que *telnet* no puede hacer una conexión remota. Las tres causas más comunes son:

- ↳ Se ha escrito mal el nombre del host.
- ↳ El host remoto no está disponible temporalmente.
- ↳ Se ha especificado el nombre de una computadora que no está en **INTERNET**.

Una vez que *telnet* hace la conexión, podemos trabajar con el host remoto, además una vez que entramos, la mayoría de los hosts muestran algún tipo de mensaje informativo, normalmente identificando a la computadora. En cuanto termina el mensaje informativo el sistema remoto nos pedirá nuestro nombre o identificador, acto seguido nos pedirá el password o palabra clave para dicho usuario.

En el caso de los hosts remotos que se instalan expreso para ofrecer algún tipo de servicio público no es necesario conocer la palabra clave para iniciar la sesión, es más si queremos utilizar algún servicio específico como *archie*, el identificador de usuario debe ser *archie*.

Otros hosts públicos son aún más automáticos, en cuanto hacemos *telnet* para conectarnos, el programa remoto se ejecuta, sin ser necesario iniciar la sesión.

Una vez que se termina el trabajo con la computadora remota, es necesario desconectar la sesión de la forma normal. La conexión se interrumpirá y *telnet* terminará automáticamente.

Ahora bien, si nosotros solamente iniciamos nuestro programa *telnet*, al aparecer el prompt en la pantalla es necesario introducir la instrucción *open* para establecer la conexión, posteriormente nos preguntará a dónde queremos conectarnos y nuevamente el nombre del usuario y su palabra clave.

Finalmente para terminar nuestra sesión de *telnet* escribimos la palabra *quit*.



## 🔑 PUERTOS VIRTUALES

Dentro de **INTERNET** hay muchos hosts que ofrecen acceso *telnet* para servicios públicos, y muchos de estos hosts necesitan que se especifique un número de puerto en particular cuando se realiza la conexión. El número de puerto identifica el tipo de servicio que se está solicitando. Veamos un ejemplo.

Hay un host en la Universidad de Michigan que proporciona informes meteorológicos de América y Canadá, el nombre de esta computadora es *downwing.sprl.umich.edu*. Y cuando queremos conectarnos con esta computadora, debemos especificar el puerto número 3000 para indicarle al sistema que queremos utilizar el servicio meteorológico.

*telnet downwing.sprl.umich.edu 3000*

(Es necesario dejar un espacio en blanco entre la dirección y el número de puerto.) Si se hace la conexión utilizando la orden *open* desde el indicador *telnet*, el número de puerto se escribe de forma análoga.

Cuando se conecta con un host remoto que usa un número de puerto, el programa que se quiere se ejecutará automáticamente, y no será necesario iniciar la sesión, y cuando terminemos el programa, la conexión se interrumpirá y *telnet* finalizará solo.

En la terminología informática, el término puerto hace referencia a la conexión entre dos dispositivos o sistemas, por ejemplo podemos conectar una impresora a un puerto de la computadora o en el caso de sistemas Unix, se dice que cada terminal está conectada a un puerto de la computadora.

Esta misma idea se utiliza en la terminología **INTERNET**, sabemos que **INTERNET** utiliza un protocolo llamado TCP (Transmission Control Protocol) para transferir datos de una computadora a otra, siempre que TCP conecta un host de **INTERNET** con otro, se utiliza un número de puerto para identificar el tipo de conexión. De hecho, hay una organización llamada Autoridad de Asignación de Números de **INTERNET** que mantiene la lista oficial de los números de puerto (hay un gran número), y asegura que se asignen números únicos cuando es necesario.



Por default, las conexiones normales de *telnet* se realizan utilizando el puerto número 23. En otras palabras, cuando no se indica un número de puerto, el programa *telnet* conecta automáticamente utilizando el puerto 23. Para poderse conectar con un host remoto, debemos ejecutar en dicha computadora un programa que esté preparado para establecer la comunicación sobre un puerto determinado. Cuando un programa de estos está esperando para establecer una conexión, se dice que está escuchando en ese puerto. De este modo, se puede decir que, para que un host admita conexiones *telnet*, debe tener un programa que escuche en el puerto 23. Otra manera de explicar esto es que cuando se utiliza *telnet* para hacer una conexión normal con un host remoto, conecta con ese host y comprueba si hay un programa escuchando en el puerto 23. Solamente cuando se quiere utilizar una conexión *telnet* sobre otro puerto diferente se necesita especificar un número de puerto. Muchos sistemas **INTERNET** utilizan diferentes números de puerto para ofrecer algún tipo de servicio especial.

Ahora, estos números de puerto son en realidad asignaciones lógicas que se hacen a los diferentes servicios de *telnet*, puesto que en realidad no existen 3000 puertos físicos en la máquina del ejemplo, por esta razón estos puertos reciben el nombre de puertos virtuales.

Cuando se trabaja con un host remoto, hay una forma de interrumpir el trabajo y dar órdenes directamente a *telnet*. Por ejemplo, si hubiera problemas con el host remoto, podemos interrumpir la sesión de trabajo, volver a *telnet* y pulsar la orden, *quit*. Para poder interrumpir la sesión de trabajo se utiliza un caracter de escape, generalmente es CTRL y el caracter de cerrar corchete '^]'. Aunque cuando iniciamos nuestra sesión aparece la secuencia de escape propia de cada sistema. Una vez interrumpida la sesión podemos teclear cualquier orden que queramos.

En ocasiones con algunas órdenes, *telnet* automáticamente continuará con la conexión después de ejecutarlas, mientras que en otros casos, se puede continuar con la conexión pulsando RETURN en el indicativo *telnet*.



## INSTRUCCIONES DE *telnet*

Para ver un resumen de las órdenes de *telnet*, se puede teclear el carácter `?`:

Este es un resumen típico:

### Instrucciones *telnet*

Commands may be abbreviated	Commands are:
Close	Close current connection
Display	Display operating parameters
Mode	Try to enter nine-by-line or character-at-a-time mode
Open	Connect to a site
Quit	Exit <i>telnet</i>
Send	Transmit special characters ('send?' for more)
Set	Set operating parameters ('set?' for more)
Status	Print status information
Toggle	Toggle operating parameters ('toggle?' for more)
z	Suspend <i>telnet</i>
?	Print help information

De todas estas órdenes, las más importantes son: `?`, `open`, `close`, `quit` y `z`.

La orden *open* indica a *telnet* que haga una conexión con la computadora remota. La orden *close* finaliza una conexión remota sin terminar el programa *telnet*. La orden *quit* finaliza el programa *telnet*. Si hay una conexión remota activa, *telnet* la terminará.

Antes de cerrar una conexión o salir de *telnet*, debe cerrarse la sesión de trabajo del host remoto. Muchos hosts desconectan la sesión cuando la conexión se interrumpe, pero es mejor asegurarse. De esta forma, se asegura que cualquier programa que se estuviera utilizando se termina de forma apropiada y todos los datos son almacenados.

Los sistemas Unix proporcionan una facilidad llamada control de trabajos que nos permiten interrumpir un programa, trabajar con otro programa y luego continuar trabajando con el primer programa. En el indicativo de *telnet*, se puede teclear la orden `z`. Esto le indica a *telnet* que interrumpa el programa para volver al shell (el programa que lee y procesa las órdenes). Esto permite que se den órdenes en la forma normal en medio de una sesión remota.



La orden `z` sólo funcionará si el shell proporciona control de trabajos. Este será el caso si se utiliza un intérprete de órdenes del tipo C-Shell o Korn shell. Si se utiliza un intérprete de órdenes más antiguo del tipo Bourne shell que no tiene control de trabajos, la orden `z` no funcionará, de hecho, *telnet* se bloqueará.

Cuando se teclea la orden `z`, *telnet* se pondrá en segundo plano y nos llevará al shell del sistema, ahí podemos todas las órdenes de Unix que queramos, por ejemplo podemos comprobar el correo electrónico, ver la fecha y la hora o cualquier otra cosa. Cuando se desee volver a la conexión remota, tenemos que escribir la orden `fg`, la cual le indica a Unix que vuelva a activar (llevar a primer plano) el último programa que se puso en segundo plano y podemos continuar con la sesión remota.

```
% telnet lpi.jsc.nasa.gov
Trying 192.101.147.11 ...
Connected to lpi.jsc.nasa.gov.
Escape character is '^['.
```

VAX/VMS V5.4

Username: LPI

VAX/VMS V5.4

Last interactive log in on Wednesday, 1-DEC-1993 12:28  
Last non-interactive log in on Wednesday, 1-DEC-1993 06:35  
LUNAR AND PLANETARY INSTITUTE MAIN MENU

- x General Information
- x Information and Research Services
- x Lunar and Planetary Bibliography
- x Image Retrieval and Processing System (IRPS)
- x Meeting Information and Abstracts
- x Venus and Mars: Gravity and Geophysics
- x Mars Exploration Bulletin Board
- x Lunar and Planetary Information Bulletin
- x Help
- x Exit and Logout





## **ftp: TRANSFERENCIA DE ARCHIVOS**

### INTRODUCCION

Uno de los servicios de **INTERNET** más importantes y utilizados es el servicio *ftp anónimo*, ya que éste nos permite copiar archivos de miles de computadoras diferentes de todas las partes de **INTERNET**. Estos archivos contienen todo tipo de información que se puede almacenar en una computadora.

El término *ftp* proviene de Protocolo de Transferencia de Archivos (File Transfer Protocol), el conjunto de especificaciones que proporcionan la transferencia de archivos en **INTERNET**. *ftp* es un servicio que permite copiar un archivo desde cualquier host de **INTERNET** a otro host de **INTERNET**. Como la mayoría de los servicios de **INTERNET**, *ftp* utiliza la arquitectura cliente/servidor, utilizando un programa cliente, denominado *ftp*, es posible conectar con un programa servidor en la computadora remota. Conceptualmente, la idea es simple. utilizando un programa cliente nuestras órdenes se envían al servidor, quien responde ejecutando las órdenes que se le envían. Por ejemplo, se puede dar una orden que solicite al servidor que envíe una copia de un archivo, el servidor responde enviando el archivo, el programa cliente recibe el archivo y lo almacena en un directorio. Cuando se copian archivos de una computadora remota a nuestra computadora, se dice que se estamos cargando un archivo, mientras que si copiamos un archivo desde nuestra computadora a otra computadora remota, se dice que estamos descargando un archivo. Para recordar la diferencia entre cargar y descargar, imaginemos que el host remoto es un almacén y nuestra computadora es un camión. Nosotros cargamos archivos en nuestro camión y descargamos archivos en el almacén.

Cuando hablemos de *ftp*, se utilizará la misma terminología que para *telnet*, nuestra computadora se denomina host local y la otra computadora host remoto. De este modo, en el lenguaje de **INTERNET**, se dice que el programa cliente de *ftp* permite cargar y descargar archivos desde y hacia un host remoto. Sin embargo, existe una restricción: no se puede acceder a una computadora a menos que se tenga la autorización apropiada. En la práctica, esto significa que se pueda iniciar una sesión con esa computadora. En otras palabras, no se pueden copiar archivos hacia o desde una computadora si no se posee un identificador de usuario (nombre de cuenta) y una palabra clave.



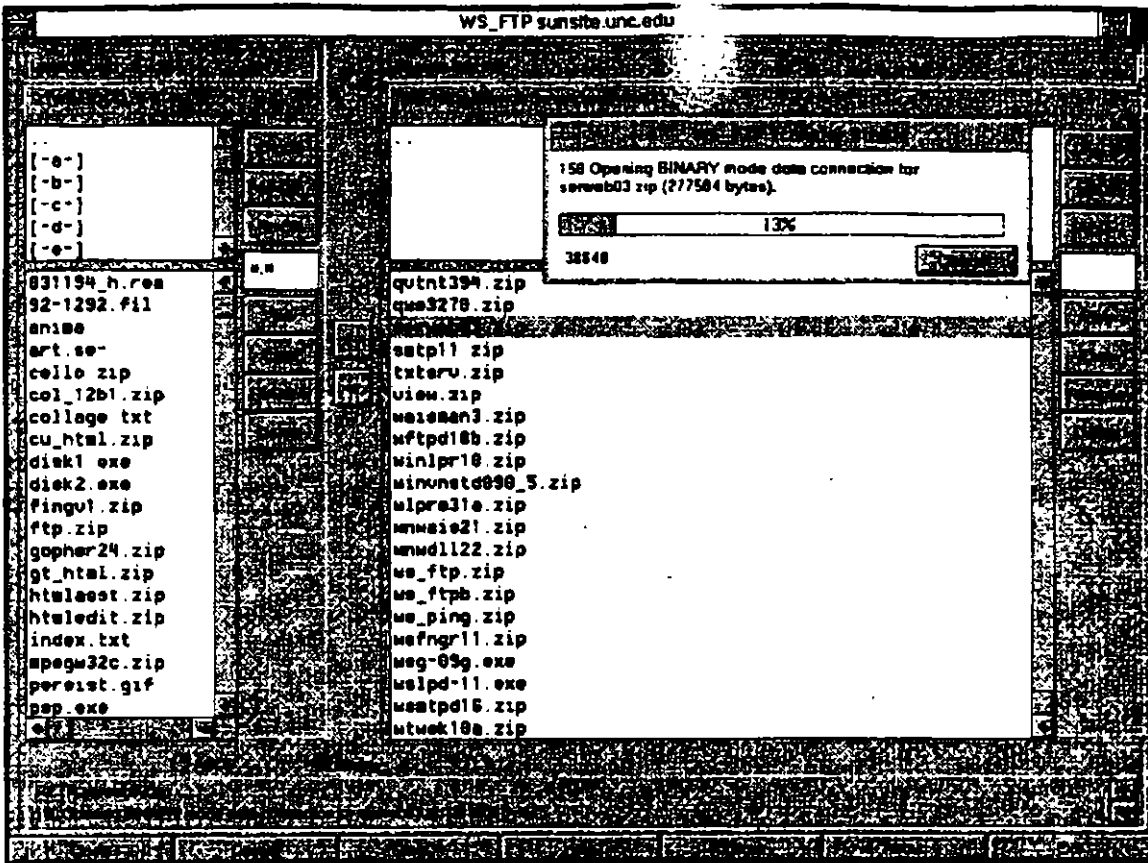
*ftp anónimo* es una facilidad que permite conectar con un host remoto y copiar archivos sin necesidad de estar registrado como usuario. El administrador del sistema configura una cuenta especial denominada *anonymous* que cualquier usuario, de cualquier lugar de la **INTERNET**, puede utilizar.

Cuando utilizamos el programa *ftp* para conectar con un servicio *ftp anónimo*, se procede en la forma normal, excepto que cuando se pide el identificador de usuario hay que escribir *anonymous* y cuando se solicita la palabra clave, hay que escribir nuestra dirección de correo, esto permite a los administradores del sistema comprobar quien está accedendo a sus archivos.

Dentro del mundo de información al que tenemos acceso, de lo más importante está la información técnica que define a la **INTERNET** misma. Ya que la **INTERNET** está basada en un gran número de protocolos y convenciones, cada uno de estos protocolos se explica en una publicación técnica llamada RFC (Request For Comments), es decir un RFC es una explicación técnica de algo que se supone que funciona. Los RFC también se utilizan para suministrar información útil en un tema específico.

Cada RFC se define con un número y se distribuye gratuitamente a cualquier usuario que quiera leerlo, por ejemplo, el RFC 1325 es una larga lista de respuestas a preguntas que son muy comunes entre los nuevos usuarios de Internet.





## EL PROGRAMA *ftp*

El programa *ftp* actúa como un cliente y nos conecta con el servidor de *ftp* en un host remoto. Una vez que se establece la conexión, al igual que en *telnet* es necesario introducir un identificador de usuario y una palabra clave (password), en el caso de que nos conectemos a un *ftp* *anónimo*, ya mencionamos cuales son las palabras adecuadas y después se pueden escribir las órdenes de *ftp* que se quieran.

Para utilizar el programa *ftp*, se escribe la orden *ftp* seguida de la dirección del host remoto con el que se quiere conectar. Una vez más podemos indicar la dirección del host remoto (su nombre) o la dirección IP del mismo. Una vez que se ejecuta el programa, se puede escribir una orden tras otra, para copiar archivos en cualquier dirección. Cuando se utiliza *ftp*, existe un gran número de órdenes posibles. Por ejemplo, se pueden explorar los directorios de la computadora remota y cambiar de un directorio a otro.



Una vez que se establece la conexión -lo que puede llevar algunos segundos si el host está muy distante- aparece un mensaje que nos indica en su primera línea que se ha establecido la conexión y aparece el alias de la computadora a la cual nos hemos conectado. Es común ver nombres que empiezan con *ftp*. La segunda línea nos informa sobre el nombre del servidor *ftp* y la versión del software de *ftp* que se está utilizando. Y finalmente la última línea nos pregunta nuestro nombre de usuario.

Si el servidor de *ftp* ha aceptado el identificador de usuario (generalmente será *anonymous*), hay que escribir nuestro password, si entramos como *anonymous* hay que escribir la dirección de nuestro correo electrónico como palabra clave.

Lo siguiente que aparece es el prompt de *ftp* y en este momento estamos listos para trabajar y ejecutar instrucciones *ftp*.

En el caso de que el programa cliente de *ftp* no puede conectar con el host remoto, nos aparecerá un mensaje indicando que o bien el host remoto está fuera de servicio, o que se interrumpió la comunicación por algún problema telefónico o que nos equivocamos al escribir el nombre.

Hay dos formas de ejecutar el programa *ftp*, una es escribir la instrucción *ftp* seguida de la dirección del host remoto, y la otra es escribir la orden por si sola y escribir después la instrucción open para abrir la conexión. Una vez que terminamos nuestro trabajo con *ftp* es importante terminar la conexión con la instrucción *quit*.



## COMANDOS DE *ftp*

En cualquier momento, es posible visualizar una lista de todas las órdenes *ftp* tecleando *?* o *help*. Algunas de las instrucciones más importantes se enlistan a continuación.

### Comandos de *ftp*

!	crmac	defproxy	send	port
\$	delete	mdelete	put	status
account	debug	mdir	pwd	struct
append	dir	mget	quit	sunique
ascii	disconnect	mkdir	quote	tenex
bell 8	form	mls	recv	trace
binary	get	mode	remotehelp	type
bye	glob	mput	rename	user
case	hash	nmap	reset	verbose
cd	help	ntrans	rmdir	?
cdup	lcd	open	runique	
close	ls	prompt	send	

Las órdenes de *ftp* se pueden dividir en varios grupos, en seguida mostramos un resumen de las órdenes más comunes.

### Comandos *ftp*

Ordenes básicas	
quit	Cerrar la conexión con el host remoto, terminar programa ftp
?	ver una lista de órdenes ftp
? orden	ver ayuda sobre la orden especificada
help	ver una lista de órdenes ftp
help orden	ver ayuda sobre la orden especificada
!	local: interrumpe ftp y ejecuta el intérprete de órdenes
! orden	local: ejecuta la orden especificada



Ordenes básicas:	
quit	cerrar la conexión con el host remoto, terminar programa <i>ftp</i>
?	ver una lista de órdenes <i>ftp</i>
? orden	ver ayuda sobre la orden especificada
help	ver una lista de órdenes <i>ftp</i>
help orden	ver ayuda sobre la orden especificada
!	local: interrumpe <i>ftp</i> y ejecuta el intérprete de órdenes
! orden	local: ejecuta la orden especificada

Ordenes de conexión	
open [host]	establecer una conexión con el host especificado
close	cerrar la conexión con el host remoto, continuar en <i>ftp</i>
user	indicar identificador usuario

Ordenes de directorios	
cd [directorio] remoto:	cambiar al directorio especificado
cdupremoto:	cambiar al directorio padre
dir [directorio[archivo]] remoto:	listar un directorio (formato largo)
lcd [directorio]local:	cambiar de directorio
ls [directorio [archivo]]remoto:	listar un directorio (formato corto)
pwdremoto:	ver el nombre del directorio actual

Ordenes de transferencia de archivos	
get [ar.remoto[ar.local]	cargar un archivo
mget [ar.remoto ... ]	cargar múltiples archivos

Configuración de opciones	
ascii	(default) definir el tipo del archivo como texto ASCII
binary	definir el tipo del archivo como binario
hash	si/no: mostrar # por cada bloque de datos transferido
pro-pt	si/no: preguntar por transferencias de archivos múltiples
status	ver el estado actual de las opciones



En el listado siguiente aparece una sesión típica con *ftp anónimo*, las órdenes que se escriben aparecen resaltadas.

```
% ftp rtfm.mit.edu  
Connected to CHARON.MIT.EDU.220 charon ftp server  
(Version 6.6 Wed Apr 14 21:00:27 EDT 1994) ready.  
  
Name (rtfm.mit.edu:pedro): anonymous  
331 Guest login ok, send e-mail address as password.  
Password: 230 Guest login ok, access restrictions apply.  
  
ftp> cd /pub/usenet/news.answers  
250 CWD command successful.  
  
ftp> hash  
Hash mark printing on (8192 bytes/hash mark).  
  
ftp> get folklore-faq  
200 PORT command successful.  
150 Opening ASCII mode data connection for folklore-faq (84701 bytes).  
#####  
226 Transfer complete.local: folklore-faq remote: folklore-faq 86113 bytes  
received in 17 seconds (4.9 Kbytes/s)  
  
ftp> quit  
221 Goodbye.
```

#### ✓ EL PROGRAMA *ftpmail*

Si no tenemos acceso a un programa *ftp*, hay una forma de realizar peticiones de servicio *ftp anónimo* por correo electrónico, para ello necesitamos utilizar un programa llamado *ftpmail*.

De igual forma si se utiliza un programa *ftp*, ocasionalmente puede ser conveniente hacer peticiones *ftp* por correo, por ejemplo cuando se solicita un listado de un directorio muy largo, también a veces es más fácil enviar una petición por correo que utilizar el programa *ftp* y hacer el trabajo uno mismo.



Para utilizar *ftpmail*, debe enviarse un mensaje a un servidor de *ftpmail*. Existen varios servidores de *ftpmail* en **INTERNET**, el principal de ellos es un host llamado *ftpmail.decwrl.dec.com*. Por lo que el mensaje hay que enviarlo a *ftpmail@decwrl.dec.com*. Dentro del mensaje necesitamos incluir órdenes una por línea, estas órdenes son instrucciones para conducir una sesión *ftp anónimo*. El servidor de *ftpmail* recibirá el mensaje y llevará a cabo las instrucciones. Los resultados de la sesión se enviarán automáticamente por correo al usuario que hizo la petición.

Además, *ftpmail* enviará una nota cuando reciba la petición y otra informando sobre la sesión de *ftp anónimo*. Cuando enviamos un mensaje a *ftpmail*, el servidor ignora el contenido de la línea de asunto, sin embargo, si se especifica algo en esta línea, *ftpmail* la incluirá en la línea de asunto cuando remita una respuesta. Por esta razón, si se envía más de una petición a *ftpmail*, es muy recomendable utilizar diferentes contenidos para esa línea que ayuden a identificar las respuestas.

Para obtener instrucciones de cómo utilizar *ftpmail*, debe enviarse una petición a *ftpmail@decwrl.dec.com*. Y como ejemplo incluimos una petición creada con *mail* de Unix.

```
% mail ftpmail@decwrl.dec.com
Subject: Solicitud de ayuda
help
CTRL-D
EOT
```

Antes de utilizar *ftpmail*, conviene leer todas las instrucciones, anexamos un resumen de las intrucciones de *ftpmail* como referencia. El principio de estas instrucciones es el mismo que el que se utiliza para *ftp anónimo*.





## Instrucciones de *ftpmail*

Especificar la dirección de correo	
<i>reply dirección</i>	dirección de correo para respuesta

Conectar con un host	
<i>connect host</i>	conectar con la dirección especificada

Configurar opciones	
<i>ascii</i>	los archivos a enviar son archivos de texto
<i>binary</i>	los archivos a enviar son archivos binarios
<i>uuencode</i>	convertir archivos binarios a texto utilizando uuencode
<i>btoa</i>	convertir archivos binarios a texto utilizando btoa

Especificar directorios	
<i>chdir directorio</i>	cambiar al directorio especificado

Solicitar archivos	
<i>get archivo</i>	enviar una copia del archivo especificado

Solicitar información	
<i>help</i>	enviar una descripción de uso de <i>ftpmail</i>
<i>dir [directorio]</i>	enviar un directorio completo
<i>ls [directorio]</i>	enviar un directorio breve (sólo nombres)

Terminar la sesión	
<i>quit</i>	terminar la sesión, ignorar el resto del mensaje



Si el archivo que se ha solicitado es muy grande (mayor de 64,000 caracteres), *ftpmail* lo dividirá en trozos y los enviará por separado, será entonces unir estos mensajes para crear el archivo original.

El ejemplo siguiente muestra una sesión en la que se utilizan las instrucciones necesarias para conectarnos con el host *ftp anónimo ftp.uu.net*, cambiarnos al subdirectorio */doc/literary/obi/DEC/humor* y traer el archivo *Lawyer.jokes.Z*.

```
% mail ftpmail@decwrl.dec.com
Subject: Ejemplo de transferencia binaria con uuencode
reply pedro.huerta@industry.net
connect ftp.uu.net
binary
uuencode
chdir /doc/literary/obi/DEC/humor
get Lawyer.jokes.Z
quit
CTRL-D
EOT
```

## **archie: LOCALIZACION DE ARTICULOS EN *ftp anónimo***

### INTRODUCCION

Para poder obtener un archivo vía *ftp anónimo*, es necesario conocer la ubicación del archivo, en particular, debemos conocer la dirección del servidor *ftp anónimo* y el nombre del directorio que contiene el archivo. La herramienta que nos sirve para encontrar estos datos es *archie*, el catálogo de la biblioteca más grande del mundo.

A lo largo de **INTERNET** hay un gran número de computadoras denominadas *Servidores archie*, que proporcionan un servicio muy importante: ayudan a localizar el nombre de los servidores *ftp anónimo* que contienen un determinado directorio o archivo.



Cuando se precisa encontrar un archivo o directorio por *ftp anónimo*, todo lo que hay que hacer es decirle a *archie* lo que se está buscando, *archie* buscará en su base de datos y mostrará el nombre de cada host *ftp anónimo* que almacena ese archivo o directorio además de mostrar también el directorio exacto. En consecuencia, todo lo que se necesita es hacer *ftp* a uno de esos hosts y copiar el archivo que se buscaba.

Hay tres formas de utilizar un servidor *archie*, primero, haciendo *telnet* a un servidor *archie* e iniciando una conexión con el identificador de usuario *archie*, una vez establecida la conexión, se ejecutará el programa *archie*. A este programa se le pueden dar instrucciones, una tras otra, y preguntarle sobre un archivo o directorio determinado. *archie* buscará en su base de datos y presentará los resultados. Si no estamos seguros del nombre del archivo que estamos buscando, *archie* tiene una facilidad, denominada servicio "*whatis*", que suministra descripciones de miles de programas, archivos de datos y documentos diferentes.

La segunda forma de utilizar *archie* es a través de un programa cliente *archie*, que se ejecuta en la computadora local. En este caso, le indicamos al cliente *archie* que es lo que estamos buscando. El cliente *archie* se conectará automáticamente con un servidor *archie*, se realizará la búsqueda, aceptará los resultados que le envía el servidor y los mostrará en la pantalla. Una vez que se aprende a utilizar *archie*, es mucho más fácil y rápido utilizar un cliente *archie*. De esta forma no es necesario iniciar una sesión *telnet*, ni recordar las órdenes de *archie*.

Por último, es posible enviar por correo peticiones a un servidor *archie*. El cual ejecutará la petición y nos enviará los resultados por correo.

Conceptualmente, el funcionamiento de *archie* es muy simple y determinados intervalos, unos programas especiales conectan con cada servidor *ftp anónimo* y copian un listado del directorio completo de todos los archivos públicos, estos listados se almacenan en lo que se denomina *Base de Datos de Archivos de INTERNET*. Cuando se solicita a *archie* que busque un archivo, todo lo que hace es mirar en la Base de Datos.



Los distintos servidores *archie* alrededor del mundo guardan la pista de los servidores *ftp anónimo* en una determinada porción de la **INTERNET**. Por ejemplo, los servidores *archie* Australianos guardan la información de todos los servidores *ftp anónimo* Australianos, esta información es compartida de tal modo, que los servidores *archie* se mantienen actualizados tanto como es posible. Haciendo un promedio, los servidores *ftp anónimo* se comprueban una vez por semana.

*archie* fue desarrollado originalmente como un proyecto por estudiantes y personal técnico de la Escuela Universitaria de Informática McGill de Montreal, Canadá, el software fue escrito por Alan Emtage y Bill Heelan con la ayuda de Peter Deutsh. Hoy en día, *archie* es desarrollado y mantenido por la empresa Bunyip Information Systems de Montreal, Canadá.

#### ☞ UTILIZACION DE *archie* CON *telnet*

Para utilizar un servidor *archie*, hay que hacer *telnet* e iniciar una sesión utilizando el identificador de usuario *archie* y no se necesita password.

Los servidores *archie* tienen generalmente un límite de usuarios que pueden utilizarlo simultáneamente. Si el servidor *archie* más cercano está ocupado, hay que utilizar otro servidor o esperar durante un tiempo y probar de nuevo.

Una vez iniciada la conexión con el identificador de usuario *archie*, aparecerán una serie de mensajes de bienvenida. Después aparecerá el indicativo del programa *archie*, en ese momento, *archie* está preparado para aceptar órdenes. Para finalizar *archie*, hay que escribir la orden quit. Para ver información de ayuda, existe la orden *help*.

Utilizar *archie* es un proceso que consta de tres partes. Primero, se definen una serie de parámetros (variables), para que el programa funcione de acuerdo con unas preferencias determinadas. Después se pregunta por los archivos a localizar. Y por último, se finaliza la sesión de *archie*.

Para controlar el entorno de trabajo, se cambian los valores de una serie de parámetros denominados *variables*. Cada variable tiene un nombre y un valor, es posible indicar a *archie* cómo debe funcionar cambiando el valor de una variable en particular. Por ejemplo, se puede indicar a *archie* que ordene la salida de una búsqueda definiendo la variable *sortby*. Es posible comprobar los valores actuales de todas las variables utilizando la orden *show*.



La mejor forma de comenzar una sesión *archie* es ejecutar la orden *show*, debemos examinar el valor de las variables más importantes y ver si se ajustan a nuestras necesidades. Si no es así, se pueden cambiar utilizando la orden *set*. Una vez definidas todas las variables necesarias, pueden ejecutarse las órdenes precisas para que *archie* realice la búsqueda de la información concreta.

A continuación mostramos un sumario con las instrucciones más importantes de *archie*.

### Principales funciones de *archie*

<b>Finalizar una Sesión <i>archie</i></b>	
<i>quit</i>	terminar una sesión <i>archie</i> y desconectar
<b>Realizar una Búsqueda</b>	
<i>find patrón</i>	buscar en la base de datos de <i>ftp anónimo</i>
<i>prog patrón</i>	el nombre antiguo de <i>find</i>
<i>whatis patrón</i>	buscar en la Base de Datos de Descripción de Software
<b>Mostrar Información</b>	
<i>help ?</i>	mostrar una lista de órdenes
<i>help orden</i>	mostrar ayuda para la orden especificada
<i>help set variable</i>	mostrar información de la variable especificada
<i>manpage</i>	mostrar la página de manual de <i>archie</i>
<i>motd</i>	mostrar de nuevo el mensaje del día
<i>servers</i>	mostrar una lista de los servidores <i>archie</i>
<i>version</i>	mostrar la versión de <i>archie</i>
<b>Enviar Información por Correo</b>	
<i>set mailto dirección</i>	especificar una dirección de correo
<i>mail [dirección]</i>	enviar la información de salida
<b>Mostrar el Valor de las Variables</b>	
<i>show</i>	mostrar el valor de todas las variables
<i>show variable</i>	mostrar el valor de la variable especificada



### Definir Variables Generales

set autologout <i>minutos</i>	definir tiempo inactivo antes de una desconexión automática
set maxhits <i>número</i>	definir el número máximo de elementos que hay
set pager	mostrar la salida utilizando un programa de paginación.
unset pager	mostrar la salida sin programa de paginación
set status	mostrar línea de estado durante la búsqueda
unset status	no mostrar línea de estado durante la búsqueda

### Definir el formato de salida

set outpu_format verbose	mostrar la salida utilizando un formato largo
set outpu_format terse	mostrar la salida utilizando un formato corto
set outpu_format machine	mostrar la salida utilizando un formato máquina
set sortby none	no ordenar la salida
set sortby filename	ordenar alfabéticamente por nombre de archivo
set sortby hostname	ordenar alfabéticamente por nombre de host
set sortby size	ordenar por tamaño, de mayor a menor
set sortby time	ordenar por hora y día
set sortby rfilename	ordenar alfabétic. por nombre de archivo (orden inverso)
set sortby rhostname	ordenar alfabétic. por nombre de host (orden inverso)
set sortby rsize	ordenar por tamaño, de menor a mayor
set sortby rtime	ordenar por hora y día (orden inverso)



## Definir preferencias de búsqueda

set search	exact buscar por un patrón exacto
set search sub	buscar todo lo que contenga un patrón igual a sub, pero distingue mayúsculas y minúsculas
set search subcase	
set search regex	buscar una expresión regular
set search exact-sub	probar exact, después sub
set search exact subcase	probar exact, después subcase
set search exact-regex	probar exact, después regex

The screenshot shows a Netscape browser window titled "NCSA Mosaic (or MS Windows)". The address bar displays "http://www.lerc.nasa.gov/Doc/archieplex.html". The main content area features the "ArchiePlexForm" search interface. It includes a search input field with the text "DI1", a dropdown menu for search types (currently set to "Case Insensitive Substring Match"), radio buttons for sorting results by "Host" (selected) or "Date", a dropdown for search impact (set to "very Nice"), a dropdown for Archie servers (set to "University of Nebraska"), a domain restriction field (empty), and a results limit field (set to "50"). There are "Submit" and "Reset" buttons at the bottom of the form.



## UTILIZACION DE LAS BASES DE DATOS *WHATIS*

Si se conoce el nombre de un archivo o directorio (o parte del nombre), puede utilizarse la orden *find* para realizar una búsqueda con *archie* en la Base de Datos de Archivos de **INTERNET**. Sin embargo, ¿que ocurre si se sabe lo que se quiere buscar, pero no se conoce su nombre?.

Si no conocemos el nombre de lo que queremos buscar, *archie* mantiene una segunda colección de información denominada la *Base de Datos de Descripción de Software*. Esta base de datos contiene descripciones breves de miles de programas, documentos y archivos de datos que se encuentran en los servidores *ftp anónimo*.

Para realizar búsquedas en esta base de datos se utiliza la orden *whatis*. Hay que escribir la orden seguida de una palabra, *archie* buscará en la Base de Datos de Descripción del Software y mostrará todos los registros que contengan la palabra especificada. Cada registro contiene una breve descripción y un nombre de archivo. Si la descripción representa la información que se busca, podrá utilizarse el nombre del archivo indicado para encontrarlo con *archie*.

La información que despliega *whatis* proviene de la información que un usuario envía a los administradores de *archie* cuando comparte un archivo vía *ftp anónimo*. Si el usuario que creó el archivo que se está buscando no envía la descripción no se encontrará en la Base de Datos de Descripción del Software. Además, al contrario que en la Base de Datos de Archivos de **INTERNET**, la Base de Datos de Descripción del Software no se actualiza de forma regular.

## CLIENTES *archie*

Un Cliente *archie* es un programa que se ejecuta en la computadora local, a este programa se le indica que información buscar, establece una conexión con un servidor *archie*, solicita la información, acepta la respuesta y la muestra en la pantalla de la computadora local.





Utilizar un cliente *archie* es muy sencillo, y además puede ser extremadamente rápido, la mayoría de las veces, al utilizar un cliente *archie*, se obtendrá una respuesta en pocos segundos. Otra consideración a tener en cuenta es que la mayoría de los servidores *archie* tienen un límite máximo de usuarios *archie* conectados al mismo tiempo y cuando se sobrepasa este límite no se pueden establecer más conexiones. Con los clientes *archie* no existe este problema, se ejecutan en la computadora local y siempre está disponible.

Existen varios tipos de clientes *archie*, los dos más conocidos son *archie* (para terminales Unix) y *xarchie* para terminales X-Window. También pueden encontrarse clientes *archie* para computadoras NeXT, PC ejecutando DOS y computadoras VAX ejecutando VMS.

Mientras se consulta sobre *archie*, es posible ver referencias a *Prospero*. Esta es una herramienta que permite acceder a información distribuida por **INTERNET**. Utilizando *Prospero*, se pueden organizar y utilizar archivos que están almacenados en computadoras remotas tan fácilmente como si estuvieran en la computadora local.

El sistema de archivos distribuidos que crea *Prospero* está basado en una idea llamada El Modelo de Sistema Virtual, muy utilizado en **INTERNET**. Cualquier usuario puede usar *Prospero* instalando un cliente *Prospero* en su computadora, los clientes y servidores *archie* utilizan la tecnología *Prospero* para implementar la relación cliente/servidor, aunque los detalles no se ven. *Prospero* fue desarrollado por Cliff Neuman del Instituto de Ciencias de la Información, una parte de la Universidad de Southern California. Para más información, se puede hacer *ftp* a *prospero.isi.edu*. También es posible subscribirse a una lista general de correo sobre *Prospero* enviando un mensaje a *prospero-request@isi.edu*.

La forma más fácil de utilizar *archie* es especificar el nombre de un archivo o directorio que se quiere buscar. Una de las ventajas de utilizar un cliente *archie* es que es posible manipular la salida utilizando las herramientas estándar de Unix. Y al igual que muchas otras órdenes de Unix, *archie* puede utilizarse con una serie de *opciones* o *conmutadores* después del nombre de la orden, con lo cual podemos ordenar la salida o realizar búsquedas combinadas tal y como lo hacemos con un *archie telnet*.



## ☞ *archie* POR CORREO ELECTRONICO

Utilizar un servidor *archie* por correo electrónico puede ser muy útil si no se tiene acceso a un cliente y por alguna razón, no se puede hacer *telnet* a un servidor *archie*. También puede ser útil cuando no se precisa una respuesta rápida o cuando se quieren realizar múltiples búsquedas, se puede enviar una petición por correo y obtener los resultados posteriormente en un mensaje.

Para utilizar *archie* por correo, hay que enviar un mensaje al identificador de usuario *archie* a la dirección de uno de los servidores *archie* como por ejemplo *archie.uqam.ca*, cuando se redacta el mensaje se puede dejar el asunto en blanco. Dentro del cuerpo del mensaje, se pueden utilizar tantas órdenes como se precisen, cada una en su propia línea.

*archie* ejecutará las órdenes y enviará una respuesta por correo. Las órdenes que pueden utilizarse son las mismas que se han descrito en la sección de *archie* con *telnet*. La única excepción es que no pueden utilizarse órdenes que no tengan sentido en una petición por correo como por ejemplo, *set pager*.

La primera orden deberá ser siempre *set mailto*, con el objeto de asegurar que *archie* conoce la dirección correcta donde enviar la respuesta. Si no se incluye esta orden, *archie* buscará la dirección de vuelta en la cabecera del mensaje. En la mayoría de las ocasiones esto funcionará, pero es mucho mejor especificar una dirección explícitamente para evitar equívocos. Por conveniencia, se puede usar la versión antigua de esta orden, que es *path*. Por tanto, las demás intrucciones son equivalentes.

La última orden que hay que utilizar en el mensaje es *quit*. Esto le indica a *archie* que el mensaje ha terminado y que ignore el resto de las líneas, ésto es muy importante, ya que algunos programas de correo permiten definir una signatura que se unirá de forma automática a todos los mensajes. Utilizando *quit*, se garantiza que *archie* ignorará todas las líneas que aparezcan después. Antes de enviar por correo una petición a un servidor *archie*, debería enviarse un mensaje solicitando información de ayuda. Esto nos permitirá conocer todas las órdenes que reconoce el servidor *archie*. En el ejemplo siguiente lo hacemos hacia *archie.rutgers.edu*.



```
% mail archie@archie.rutgers.edu  
Subject:  
set mailto pedro.huerta@industry.net  
help  
quit  
CTRL-D  
EOT
```

Cuando se reciba la respuesta, se debe leer cuidadosamente. Si el servidor *archie* utiliza una versión antigua del software *archie*, no se podrán utilizar todas las órdenes descritas anteriormente, en el siguiente ejemplo utilizaremos instrucciones anteriores para buscar el archivo SEI.DAT:

```
% mail archie@archie.rutgers.edu  
Subject:  
set mailto pedro.huerta@industry.net  
set maxhits 25  
set output_format verbose  
set search exact  
set sortby time  
find SEI.DAT  
quit  
CTRL-D  
EOT
```

Una vez que se reciba la respuesta del servidor *archie*, se puede enviar por correo una petición a un servidor *ftp anónimo* con el fin de obtener el archivo.

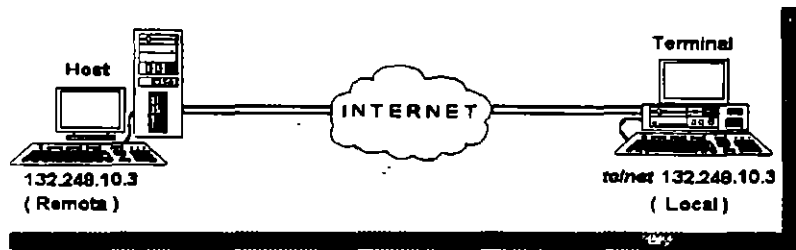


# INTERNET



## Sesión Remota Interactiva

*telnet* : Emulación de terminal



Notas:

# INTERNET

---



## ***telnet* : Puertos Virtuales**

***telnet* 132.248.10.3 23**



**Puerto virtual**

***telnet* downwing.sprl.umich.edu 3000**

↳ **Se utiliza para servicios especiales.**

↳ **El default es el puerto 23.**

**Notas:**

# INTERNET



**telnet :**

```
% telnet lpi.jsc.nasa.gov
Trying 192.101.147.11 ...
Connected to lpi.jsc.nasa.gov.
Escape character is '^]'.

```

VAX/VMS V5.4

Username: LPI

VAX/VMS V5.4

Last interactive log in on Wednesday, 1-DEC-1993 12:28  
Last non-interactive log in on Wednesday, 1-DEC-1993 08:35  
LUNAR AND PLANETARY INSTITUTE MAIN MENU

- x General Information
- x Information and Research Services
- x Lunar and Planetary Bibliography
- x Image Retrieval and Processing System (IRPS)
- x Meeting Information and Abstracts
- x Venus and Mars: Gravity and Geophysics
- x Mars Exploration Bulletin Board
- x Lunar and Planetary Information Bulletin
- x Help
- x Exit and Logout

**Notas:**

# INTERNET

---



## Transferencia de archivos

**ftp:** file transfer protocol

- ↳ Permite copiar un archivo desde cualquier host Internet a otro host Internet.
- ↳ Uno de los servicios más importantes de Internet.
- ↳ Arquitectura cliente/servidor.
- ↳ Usuario general : *anonymous*.
- ↳ Clave de acceso: dirección de correo electrónico.
- ↳ También se le conoce como *ftp anónimo*.
- ↳ Existe la opción *ftp mail*.

Notas:

# INTERNET



## *ftp, ejemplo*

```
% ftp rtfm.mit.edu
Connected to CHARON.MIT.EDU.220 charon  ftp server
(Versión 6.6 Web Arp 14 21:00:27 EDT 1994) ready.

Name (rtfm.mit.edu:pedro): anonymous
331 Guest login ok, send e-mail address as password.
Password: 230 Guest login ok, access restrictions apply.

ftp> cd / pub / usenet / news.answers
250 CWD command successful.

ftp> hash
Hash mark printing on (8192 bytes/hash mark).

ftp> get folklore-faq
200 PORT command successful.
150 Opening ASCII mode data connection for folklore-faq (84701 bytes).
#####
226 Transfer complete.local: folklore-fag remote: folklore-faq 88113 bytes
received in 17 seconds (4.9 Kbytes/s).

ftp> quit
221 Goodbye.
```

Notas:



# INTERNET

---



## *ftp, ejemplo*

```
% mail ftpmail@decwrl.dec.com
Subjet: Ejemplo de transferencia binaria con uuencode
reply pedro.huerta@industry.net
connect ftp.uu.net
binary
uuencode
chdir / doc / literary / obi / DEC / humor
get Lawyer.jokes.Z
quit
CTRL-D
EOT
```

Notas:

# INTERNET

---



## ***Localización de artículos en f t p***

### ***archie:***

- ↳ Herramienta de búsqueda de archivos.
- ↳ Permite acceder el catálogo de la biblioteca de archivos más grande del mundo.
- ↳ Existen servidores y clientes *archie*.

**Notas:**

# INTERNE



**archie:**

The screenshot shows a web browser window with the title "ArchiePlexForm". The page content includes:

- A header section with the title "ArchiePlexForm".
- A paragraph: "This is a HTML based version of ArchiePlex, an Archie gateway for the WWW. See also [documentation on ArchiePlex](#). Please remember that Archie searches can take a long time. Tip: store [this document](#) on your hard for future access! You need a [Form Browser](#) to use this. (If you haven't see [ARCHIEFORM](#))"
- A search input field: "What would you like to search for?" with a text box containing "xxx".
- A section for search options: "There are several types of search: [Name](#) [Country](#) [IP](#) [Submitting Sites".](#)
- A section for result sorting: "The results can be sorted  By Host or  By Date".
- A section for result limits: "The impact on other users can be:  Many Sites ".
- A section for server selection: "Several Archie Servers can be used: [Country](#) or [Region](#)".
- A section for domain restriction: "You can restrict the results to a domain (e.g. 'gov'):" with a text box.
- A section for result count: "You can restrict the number of results (default 50):" with a text box containing "50".
- A "Press this button to submit the query:" section with a "Submit" button.
- A "To reset the form, press this button:" section with a "Reset" button.
- A "Go/Next" button at the bottom.

**Notas:**

# INTERNET

---



## ***archie:***

### **Existen tres formas de utilizarlo**

- ↳ *telnet* a un servidor *archie*.  
Identificador de usuario: *archie*.
  - Se puede utilizar el servicio "*whatis*".
- ↳ Utilizar un cliente *archie*.
- ↳ Enviar por correo peticiones a un servidor *archie*.

**Notas:**

# CURSO: **I N T E R N E T**

5.- gopher, veronica y jughead



Octubre de 1995.

## 5.- *gopher*, *veronica* y *jughead*

### INTRODUCCION

*gopher* es un potente sistema que permite acceder a muchos de los recursos de **INTERNET** de una forma simple y consistente. Para usar *gopher*, todo lo que se necesita es seleccionar en un menú y cada vez que se hace una selección, *gopher* hace lo necesario para llevar a cabo nuestra petición. Por ejemplo, si seleccionamos un elemento de un menú que representa un archivo de texto, *gopher* traerá el archivo (dondequiera que se encuentre) y nos lo mostrará.

Algunos elementos de los menús representan otros menús, si elegimos uno de éstos, *gopher* obtendrá el nuevo menú y nos lo mostrará. De esta manera, nos podemos mover de menú en menú, usando sólo un puñado de teclas (o un ratón) para navegar.

La potencia de *gopher* viene de que los recursos enumerados en un menú pueden estar en cualquier parte de **INTERNET**. Cuando se selecciona un elemento, *gopher* lo traerá o hará lo que sea necesario para atender nuestra petición, la mayoría del tiempo, *gopher* tendrá que conectarse a otra computadora, pero todo será transparente. Todo lo que notaremos es que nuestra petición ha sido cumplida simple y fácilmente.

*gopher* es un sistema cliente/servidor, por lo que para usar *gopher*, hay que ejecutar un programa llamado *cliente gopher*. Este programa nos muestra los menús y lleva a cabo nuestras peticiones, en ocasiones el cliente *gopher* contacta con un *servidor gopher* para pedirle información en nuestro nombre. Si fuera necesario contactar con otro tipo de servicio por ejemplo, abrir una sesión *telnet* o cargar un archivo, el cliente *gopher* también se encargará de eso.



Hay varios miles de servidores *gopher* en **INTERNET**, y almacenan todo tipo de información de tal suerte que podemos encontrar servidores *gopher* en muchas universidades, empresas y otras organizaciones, de hecho, dentro de las universidades, será frecuente encontrar diferentes departamentos que tienen su propio servidor *gopher*. Cada servidor *gopher* contiene información de interés para los usuarios locales, por ejemplo, el servidor *gopher* de un departamento universitario tendrá información de interés para los miembros de ese departamento. El servidor *gopher* de una empresa tendrá información para sus clientes y empleados.

Lo importante de este sistema es que, en su mayor parte, todos los servidores *gopher* de **INTERNET** son públicos, aunque la mayoría de los servidores *gopher* han sido instalados para un conjunto particular de personas, se puede ver que la información que se encuentra es de interés general.

*gopher* ofrece más información de la que una persona puede abarcar, y nos referimos a la suma total de toda esta información -todo lo que está disponible vía *gopher*- como *gopherespacio*. Para ayudarnos a percibir lo que es viajar por el *gopherespacio*, y lo único que tenemos que hacer para navegar por él, es teclear o seleccionar las diferentes opciones del menú de *gopher*.

El *gopher* original fué desarrollado en la Universidad de Minnesota en Abril de 1991 por un equipo formado por Bob Alberti, Farhad Anklesaria, Paul Lindler, Mark McCahill y Daniel Torrey. El trabajo se hizo en el Departamento de Computadoras y Servicios de Información para proporcionar a varios departamentos del campus una forma fácil y barata de poner información disponible a todo el campus. La idea fué -y todavía es- que cada organización interesada pueda mantener su propio servidor *gopher* y poner lo que quiera en él, de esta forma, cada organización tiene control sobre su propia sección del *gopherespacio*, que comparte con cualquiera que tenga un cliente *gopher*.



## CLIENTES *gopher*

Hay una gran variedad de clientes *gopher* disponibles para muchos sistemas. Por una parte existen los clientes que se utilizan de acuerdo a la plataforma sobre la cual esté nuestro sistema, y por la otra aquellos que se pueden ejecutar desde los servidores de clientes *gopher* de acceso público, de los cuales hablaremos más tarde. El cliente *gopher* más comúnmente utilizado es el que se ejecuta en los sistemas Unix para terminales de texto, el nombre de este programa es *gopher*. Hay también un cliente *gopher*, llamado *xgopher*, para el sistema X-Window. Si se puede elegir, hay que usar un cliente *gopher* que obtenga todas las ventajas de las características especiales de nuestra computadora, es decir, si nuestro sistema utiliza menús desplegables, barras de desplazamiento y un ratón, un cliente *gopher* configurado para nuestra computadora hará uso de estas características.

Cada vez que viajamos por el *gopherespacio* nos adentramos más y más en los diferentes menús que nos presenta *gopher*, afortunadamente existe una tecla dentro de los clientes *gopher* que nos permite seleccionar las diferentes opciones y con ello generar nuestros propios menús, al marcar las opciones se dice que estamos generando una lista de marcas y cada una de ellas se generan en un cliente y se pueden mantener ahí.

De vez en cuando, podemos tener la dirección de un servidor *gopher* en particular que queramos usar. Si es así, se puede introducir la dirección como parte de la orden *gopher*. Esto le dice a *gopher* que conecte directamente con ese servidor específico en lugar de utilizar el servidor *gopher* por default, una vez más podemos utilizar direcciones IP o nombres.

Algunos servidores *gopher* necesitan que se especifique un número de puerto determinado, es decir que para acceder al servicio de *gopher* es necesario hacerlo a través de un puerto virtual, por ejemplo, existe un *gopher* en la Universidad de Tuebingen en Alemania que contiene documentos sobre MUD (realidad virtual multiusuario). La dirección se lista como *nova.tat.physikuni-tuebingen.de 4242*. El número 4242 es el número de puerto, y simplemente hay que escribirlo como parte de la orden.



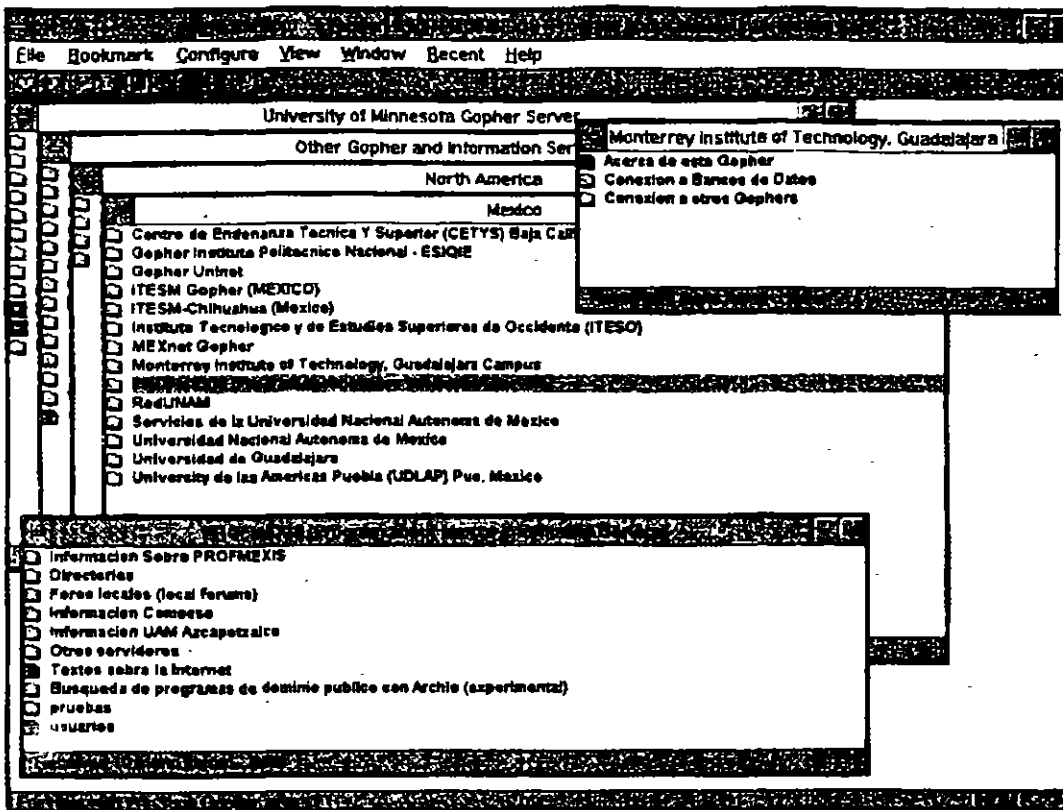


Una vez ejecutado *gopher*, nuestro cliente contactará con el servidor *gopher* que se haya designado como lugar para empezar y pedirá el menú principal, si nuestra organización tiene su propio servidor *gopher*, éste, sin duda, será inicial. En cualquier otro caso, el administrador del sistema habrá elegido algún servidor *gopher*, en alguna parte de **INTERNET**, como punto de entrada. En cuanto veamos el menú principal, ya estaremos preparados para comenzar: simplemente hay que hacer la primera selección.

Si queremos aprender más sobre *gopher* hay varios sitios donde buscar, primero, muchos servidores *gopher* tienen opciones en el menú que muestran información sobre el propio *gopher*. Segundo, hay dos grupos de discusión de *Usenet* dedicados a temas relacionados con *gopher*: *comp.infosystems.gopher* y *alt.gopher*.

Finalmente, se puede leer la lista de preguntas más frecuentes (FAQ) sobre *gopher*, podemos obtener este documento vía *ftp anónimo* de *rtfm.mit.edu* que se encuentra dentro del directorio */pub/usenet/news.answers* y traer el archivo llamado *gopher.faq*.





Para utilizar un cliente de acceso público, lo que necesitamos es hacer *telnet* a una de las máquinas enunciadas en la tabla 5-1. Una vez conectados usando el identificador de usuario especificado, automáticamente se ejecutará un cliente *gopher*. En algunos casos al utilizar un cliente *gopher* de acceso público hay restricciones en lo que se puede hacer, por ejemplo, después de que *gopher* haya mostrado un archivo de texto, veremos un mensaje que nos permite enviar el archivo por correo, salvarlo o imprimirlo. Si ejecutamos un cliente *gopher* local podríamos hacer cualquiera de estas tres operaciones, mientras que en el cliente *gopher* remoto, sólo podemos utilizar la opción de enviar por correo (ya que no podemos guardar o imprimir archivos en una computadora remota).



**Tabla 5-1. Clientes *gopher* de acceso público que se pueden usar con *telnet***

Localización	Dirección INTERNET	Dirección IP	Conectarse como...
Alemania	gopher.th-darmstadt.de	130.83.55.75	gopher
Australia	info.anu.edu.au	150.203.84.20	info
Chile	gopher.puc.cl	146.155.1.16	gopher
Dinamarca	gopher.denet.cl	129.142.6.66	gopher
Ecuador	ecnet.ec	157.100.45.2	gopher
España	gopher.uv.es	147.156.1.12	gopher
Inglaterra	gopher.brad.ac.uk	143.53.2.5	info
Japón	gopher.ncc.go.jp	160.190.10.1	gopher
Suecia	gopher.chalmers.se	129.16.221.40	gopher
Suecia	gopher.sunet.se	192.36.125.2	gopher
USA:California	infopath.ucsd.edu	132.239.50.100	infopath
USA:California	scilibx.ucsc.edu	128.114.143.4	gopher
USA:Georgia	grits.valdosta.peachnet.edu	131.144.8.206	gopher
USA:Illinois	gopher.uiuc.edu	128.174.5.61	gopher
USA:Iowa	panda.uiowa.edu	128.255.40.201	-
USA:Michigan	gopher.msu.edu	35.8.2.61	gopher
USA:Minnesota	consultant.micro.umn.edu	134.84.132.4	gopher
USA:Carolina del N.	gopher.unc.edu	152.2.22.81	gopher
USA:Carolina del N.	twosocks.ces.ncsu.edu	152.1.45.21	gopher
USA:Ohio	gopher.ohiolink.edu	130.108.120.25	gopher
USA:Virginia	ecosys.drdr.virginia.edu	128.143.96.10	gopher
USA:Virginia	gopher.virginia.edu	128.143.22.36	gwis
USA:Washington	wsuaix.csc.wsu.edu	134.121.1.40	wsuinfo

### ORDENES BASICAS DE *gopher*

Hay una gran variedad de órdenes que se pueden usar para navegar por el *gopherespacio*. La mayor parte del tiempo, podemos ir a donde queremos usando tan sólo seis teclas: las cuatro teclas de desplazamiento del cursor; la BARRA ESPACIADORA; y la tecla *B*. No obstante existen tres órdenes básicas que conviene recordar siempre. Primera, para ver la información de ayuda sobre todas las órdenes, con la tecla *?* podemos obtenerla, segunda, para terminar el cliente *gopher*, presionamos la tecla *q*, con lo cual se nos preguntará si realmente queremos salir, a lo que hay que contestar y (yes) o *n* (no). Y finalmente, para salir inmediatamente, sin confirmar, necesitamos presionar la tecla *Q* ("Q" mayúscula).



*gopher* funciona en lo que se llama *modo cbreak*, lo que significa que, para las órdenes de un sólo carácter, no es necesario pulsar la tecla RETURN, por ejemplo, para terminar *gopher*, sólo es necesario pulsar la tecla *q*. No hay que pulsar RETURN.

En todo momento existe un puntero, mostrado como -->, el cual se encuentra al lado de alguna de las opciones y lo que debemos hacer es mover el puntero a la opción deseada y seleccionarla. Para seleccionar una opción, se pueden pulsar la tecla de desplazamiento de cursor a la derecha o la tecla RETURN.

Si seleccionamos otro menú, *gopher* lo traerá y nos lo mostrara, si seleccionamos un archivo de texto, *gopher* traerá una copia y nos lo mostrara página a página.

Ahora, existen varias formas de mover este puntero. La manera más fácil es usar las teclas de desplazamiento del cursor arriba y abajo. Si el teclado no tiene estas teclas, o si prefiere no usarlas, se pueden usar *k* o CTRL-P para moverse hacia arriba, y *j* o CTRL-N para moverse hacia abajo, aunque estas alternativas pueden parecer extrañas, se usan en el editor de texto *vi* y son habituales para un gran número de usuarios de Unix.

Si se quiere saltar y seleccionar una opción en particular, simplemente hay que introducir su número. Por ejemplo, para seleccionar la opción número 10, escribimos *10* y después pulsamos la tecla RETURN.

Algunos menús son tan largos que no caben en una sola página. En este caso, el mensaje de la parte inferior derecha de la pantalla nos indicará que hay más de una página. Y para moverse a la página siguiente, hay que pulsar la BARRA ESPACIADORA. Para moverse a la página anterior, hay que pulsar *b*. De nuevo, hay teclas alternativas que se pueden usar. Para moverse a la página siguiente, se puede pulsar > (mayor que) ó + (más). Para moverse a la página anterior, se puede pulsar < (menor que) ó - (menos).

Si un menú es tan largo que requiere más de una página, se puede saltar y seleccionar una opción, aunque no se encuentre en la página actual. Por ejemplo, digamos que estamos viendo la página 7 de un menú muy largo. Se puede seleccionar la opción 1 directamente. Simplemente hay que escribir *1* y pulsar RETURN.



Otra forma de saltar es decirle a *gopher* que busque en el menú actual una opción que contenga un determinado patrón. Para hacer esto, hay que escribir un carácter "/" (diagonal), seguido de un patrón y después pulsar RETURN. Cuando busca, *gopher* no distingue entre letras mayúsculas y minúsculas.

Por ejemplo, digamos que estamos viendo una lista muy larga de servidores *gopher*, y queremos saltar directamente a la opción de la Universidad de Michigan entonces introducimos */michigan* y *gopher* nos traerá alguna opción, si la opción que encuentra *gopher* no es la que deseamos, se puede repetir la misma búsqueda pulsando la tecla *n*.

Como ya sabemos, muchas de las opciones que se seleccionan son a su vez otros menús de tal suerte que cuando se seleccionan estas opciones, *gopher* pide el menú y nos lo muestra. Así, viajar por el *gopherespacio* supone moverse arriba y abajo por un árbol de menús.

Hay unas pocas órdenes que se pueden usar para moverse de un menú al siguiente. Como mencionamos anteriormente, para seleccionar un nuevo menú, se apunta a esa opción y se pulsa la tecla de desplazamiento del cursor a la derecha o RETURN, Para volver al menú anterior, se pulsa la tecla de desplazamiento de cursor a la izquierda o la tecla *u*. En cualquier momento, se puede pulsar *m* para saltar al menú principal (con el que empieza *gopher*). Esto es útil cuando estamos muy abajo en una serie de menús y queremos volver al principio.

Hay varias formas de guardar un archivo, una vez que *gopher* ha mostrado un archivo, veremos:

*Press <RETURN> to continue, <m> to mail, <s> to save, or <p> to print:*



En este punto se puede enviar a alguien una copia del archivo por correo, guardarlo o imprimirlo. Hay un atajo para guardar un archivo sin tener que visualizarlo primero. Simplemente hay que moverse a la opción del menú y pulsar *s*. Entonces *gopher* nos pedirá que introduzcamos un nombre de archivo. *gopher* nos sugerirá un nombre de archivo que podemos aceptar o modificar como deseemos. Guardar tiene dos limitaciones. Primera, la opción actual tiene que ser algo que tenga sentido guardarlo en un archivo personal. Por ejemplo, se puede guardar un archivo de texto, pero no se puede guardar un menú. Segunda, hay que usar un cliente *gopher* que se ejecute en nuestra propia computadora. Esto tiene sentido ya que no tenemos permiso para guardar archivos en una máquina remota.

Otra forma de guardar un archivo es cargándolo, lo cual es útil cuando se usa una PC o una Macintosh para conectarse a un nodo remoto de **INTERNET**. Se puede usar la orden **D** ("D" mayúscula) para cargar un elemento directamente en la computadora y veremos un menú pidiéndonos que elijamos qué protocolo de transmisión de archivos (método de carga) queremos usar y después de hacer la elección, *gopher* iniciará la transferencia del archivo. Dependiendo de como esté configurado nuestro programa de comunicaciones, se puede ver un mensaje pidiéndonos que iniciemos la transferencia. (Todo el proceso lo describiremos en el capítulo 7)

También podemos usar una marca para guardar un elemento de un menú, *gopher* almacena todas las marcas en una lista de marcas a la que se puede saltar siempre que se quiera lo cual permite recordar y visitar cualquier lugar en el *gopherespacio*. Si usamos un cliente *gopher* en nuestra propia máquina, nuestra lista de marcas se guardará automáticamente, para que podamos usarla cada vez que exploremos el *gopherespacio*.

Hay varias órdenes que se pueden usar para crear una lista personal de marcas. Para añadir un elemento de un menú a la lista, hay que moverse a él y pulsar **A** ("A" mayúscula). En cualquier momento, podemos saltar a la lista de marcas (que es en sí misma un menú) pulsando *v*. Para volver al menú anterior, simplemente hay que pulsar la tecla de desplazamiento del cursor a la izquierda o la tecla *u*. Mientras estamos viendo la lista de marcas, podemos borrar un elemento moviéndonos a él y pulsando *d*. El arte de usar bien *gopher* es construir una lista personalizada de marcas, ya que mucha gente olvida usar esta posibilidad y acaba repitiendo los mismos pasos una y otra vez.



Además, es demasiado fácil olvidar donde encontramos un elemento determinado en el *gopherespacio*. Si no guardamos los elementos interesantes en la lista de marcas, nunca los volveremos a ver.

La siguiente orden muestra información técnica sobre un elemento de un menú. Simplemente hay que moverse a él y pulsar = (signo igual) con lo que veremos una descripción técnica del elemento tal y como aparece en la base de datos del servidor *gopher*. Lo interesante, es que se puede ver donde está localizado un elemento en el *gopherespacio*.

La última orden que vamos a mencionar es O ("O" mayúscula), que le dice a *gopher* que muestre una lista de opciones, podemos examinar la lista y hacer cualquier cambio que deseemos.

A modo de sumario mostramos a continuación las órdenes básicas de *gopher*:

### Comandos de *gopher*

Órdenes básicas	
Q	salir de <i>gopher</i> inmediatamente
q	salir de <i>gopher</i> , con confirmación
?	visualizar un sumario de ayuda
O	observar información técnica sobre un elemento
=	examinar y cambiar opciones de <i>gopher</i>

Órdenes fundamentales para moverse por el <i>gopherespacio</i>	
Flecha Derecha	seleccionar el elemento actual
Flecha Izquierda	ir al menú previo retrocediendo un nivel
Flecha Arriba	mover el puntero hacia arriba un elemento
Flecha Abajo	mover el puntero hacia abajo un elemento
Barra espaciadora	mover a la siguiente página de un menú
b	mover a la página previa de un menú
número	saltar y seleccionar un elemento específico
lpatrón	buscar el siguiente elemento del menú que contenga ese patrón
n	buscar el siguiente elemento del menú del mismo patrón



Ordenes alternativas para mover el puntero	
RETURN	seleccionar el elemento actual
u	ir al menú previo retrocediendo un nivel
k	mover el puntero hacia arriba un elemento
j	mover el puntero hacia abajo un elemento
CTRL-P	mover el puntero hacia arriba un elemento
CTRL-N	mover el puntero hacia abajo un elemento
>	mover a la siguiente página de un menú
<	mover a la página previa de un menú
+	mover el puntero hacia abajo un elemento
-	mover a la página previa de un menú

Guardar información	
s	guardar el elemento actual en un archivo
D	cargar el elemento actual en un archivo

Utilización de marcar	
a	añadir el elemento actual a la lista de marcas
A	añadir el menú actual o la búsqueda a la lista de marcas
d	borrar una marca
v	saltar a la lista de marcas

## TIPOS DE RECURSOS DE *gopher*

Cuando observamos un menú de *gopher*. Cada elemento termina con un símbolo en particular, el cual nos dice qué tipo de recurso representa el elemento. El símbolo más común que veremos al final de un elemento es un carácter "/" (diagonal), lo que indica que el elemento representa otro menú, y generalmente es el menú principal de otro servidor *gopher*.

Una cosa que se puede notar es que la estructura de los menús de *gopher* se parece al sistema de archivos de Unix. Menús y sub-menús corresponden a directorios y subdirectorios, por esta razón, algunas veces veremos referencias a los menús de *gopher* como directorios. Por ejemplo, cuando estamos esperando a que aparezca un menú de un servidor *gopher* remoto, vemos el mensaje:





## *Retrieving Directory...*

De vez en cuando, se nota que *gopher* es lento, y en estos casos lo único que podemos hacer es esperar, podríamos en un momento determinado estar tentados a pulsar CTRL-C, pero todo lo que ocurrirá es que aparecerá un mensaje preguntando si queremos abortar el propio cliente *gopher*, no hay una forma simple de parar la actividad que está desarrollando *gopher*.

El siguiente símbolo más habitual es un caracter . (punto), el cual indica que el elemento es un archivo de texto o algo que puede ser visualizado. Cuando se selecciona un elemento como este, *gopher* lo muestra, página a página, si somos un usuario de Unix utilizando nuestro propio cliente *gopher*, podemos especificar el programa de paginación que queramos, para lo cual necesitamos definir en la variable de entorno PAGER el nombre de nuestro programa favorito de paginación, por ejemplo, si utilizamos C-shell, podemos anexar la siguiente orden en el archivo de inicialización *.login*:

```
setenv PAGER more
```

Si utilizamos Korn shell o la Bourne shell, podemos poner las dos ordenes siguientes en el archivo de inicialización *.profile*:

```
PAGER=more  
export PAGER
```

Esto asegurará que la variable de entorno PAGER queda debidamente definida cada vez que nos conectemos, además, la variable es usada por un gran número de programas (tales como los programas de correo electrónico) y no sólo por el cliente *gopher*.

El símbolo <TEL> indica un tipo de recurso totalmente diferente: una sesión *telnet*. Si seleccionamos este elemento, *gopher* iniciará una conexión *telnet* a un host remoto. Justo antes de hacer la conexión, veremos un mensaje diciendo que, para hacer la conexión, vamos a ir fuera del sistema *gopher*. Lo que significa que estamos embarcándonos en una sesión *telnet* normal, para continuar, hay que pulsar RETURN y para olvidarnos de todo, hay que pulsar CTRL-G.



Una vez terminada la conexión *telnet*, volveremos al cliente *gopher*, sin embargo, durante la sesión, todo lo que escribamos estará bajo los auspicios de *telnet*. En particular, se puede usar la tecla de escape de *telnet*, CTRL-].

El siguiente símbolo *gopher* es <CSO>. Este símbolo indica una utilidad llamada *servidor de nombres CSO* que contiene información sobre personas de una organización en particular. Por ejemplo:

⌘ *University of Minesota Directory* <CSO>

En este caso, se puede seleccionar este elemento para buscar información sobre alguien de la Universidad de Minnesota. Por ejemplo, podemos necesitar encontrar la dirección de correo electrónico de alguien. El nombre "CSO" hace referencia a la Oficina de Servicios Computacionales de la Universidad de Illinois, Urbana, donde fué desarrollado el software inicialmente. Hay muchos servidores de nombres CSO en **INTERNET** y es común ver uno en el menú principal del servidor *gopher* de una Universidad u otra gran organización. Los servidores de nombres CSO pertenecen a una familia de programas llamados "directorios de páginas blancas" que se usan para encontrar a alguien en la **INTERNET**. (Hablaemos de esto en el capítulo 7)

Si se selecciona el elemento <CSO>, veremos una pantalla que nos pide lo que queremos buscar, para lo cual necesitamos conocer al menos el nombre, la dirección de correo o el departamento al que corresponde la persona que andamos buscando, si usamos unos cuantos servidores de nombres CSO, encontraremos diferentes variaciones. Sin embargo, en principio, todos funcionan de la misma manera. Cada una de las categorías -name, email, y department- se llama *campo*. Todo lo que hay que hacer es llenar algún campo y pulsar RETURN. El servidor CSO comprobará su base de datos y mostrará lo que encuentre como un único archivo de texto.

El siguiente símbolo es <?>. Este indica una base de datos que se puede consultar introduciendo una o más palabras clave. Cuando seleccionamos este elemento, vemos una pantalla que nos pide que especifiquemos lo que queremos buscar, llenamos algún campo y se pulsa RETURN, *gopher* iniciará la búsqueda y una vez que obtenga los resultados, los mostrará en forma de menú. Simplemente hay que seleccionar lo que nos parezca bien.



Por supuesto, los resultados de la búsqueda serán tan buenos como lo sea la base de datos en la que estemos buscando y sólo serán tan exactos como las palabras clave que hayamos especificado. Un problema es que, a no ser que la descripción del elemento nos lo diga, no hay forma de saber que tipo de programa de búsqueda y que base de datos estamos usando. Si estamos buscando archivos, podemos buscar a través de *archie*, o si estamos buscando texto, podemos utilizar *WAIS*.

Si nos gusta como resulta una búsqueda, podemos guardarla para utilizarla de nuevo otro día, para lo cual, hay que esperar a que termine la búsqueda y usar la orden *A*. Esto guardará la búsqueda (no sus resultados) en la lista de marcas. Siempre que queramos, podemos ir a la lista de marcas y seleccionar esa búsqueda exactamente.

Los dos últimos símbolos del menú de los que vamos a hablar son: *<Picture>* y *<I>*. El símbolo *<Picture>* indica un archivo binario que contiene una imagen, mientras que el símbolo *<I>* indica un archivo binario que contiene un sonido. Sólo podemos acceder a ellos si la computadora tiene el hardware y software adecuados, dependiendo de nuestra máquina podemos usar la orden *D* para traer esos archivos a nuestra computadora. Por supuesto, nuestra computadora debe ser capaz de entender el formato particular de esos archivos que se están usando.

#### *veronica*

*veronica* es un recurso basado en *gopher* que se utiliza para buscar en el *gopherespacio* todos los elementos de los menús que contengan unas palabras determinadas. Por ejemplo, digamos que estamos interesados en chistes de matemáticas, sabemos que en algún lugar del *gopherespacio* tiene que haber algún elemento de un menú que contenga chistes de matemáticas, y *veronica* puede encontrarlos.

Accesar a *veronica* es fácil. Es simplemente un elemento del menú que se puede seleccionar como cualquier otro, generalmente la opción dice textualmente "Buscar títulos en el *gopherespacio* utilizando *veronica*". Y solamente necesitamos encontrar *veronica* una vez, puesto que la podemos añadir a la lista personal de marcas usando la orden *a*. En cualquier momento, se puede usar la orden *v* para ver esta lista y seleccionar *veronica*.



Una vez que hayamos seleccionado un elemento de un menú que hace mención a *veronica*, veremos una lista de elementos relacionados con *veronica*. Típicamente ésta será relacionada con los lugares a donde queremos buscar, como ejemplo nuestro menú puede darnos opciones de búsqueda en los servidores *gopher* de *NYSERnet*, *University of Pisa*, *CyberNet*, etc.

Una vez que seleccionamos el lugar a donde *veronica* va a realizar la búsqueda se nos pide que especifiquemos la palabra o tema relacionado con lo que queremos buscar, retomando el ejemplo anterior en el que deseamos buscar chistes de matemáticas, escribimos *math* y pulsamos RETURN.

Una vez que hemos pulsado RETURN, *veronica* buscará en todo el *gopherspacio* conocido elementos de menús que contengan la palabra *math*. Estos elementos serán seleccionados y colocados en un menú que nos será presentado. Lo bueno de *veronica* es que nos muestra sus resultados en la misma forma que un menú normal de *gopher*. Por lo tanto, se pueden usar las órdenes estándar de *gopher* para examinar los resultados de la búsqueda de *veronica*. Si nos gusta lo que vemos, podemos usar la orden *a* para añadir un elemento a nuestra lista personal de marcas. Algo que es importante es que cuando *veronica* realiza una búsqueda, no distingue entre letras mayúsculas y minúsculas.

En nuestro ejemplo, estamos buscando chistes sobre matemáticas, pero nuestra búsqueda ha sido muy general. *veronica* ha encontrado todo tipo de elementos que tienen que ver con matemáticas, pero que no son divertidos. La mejor idea es realizar otra búsqueda, esta vez utilizando dos palabras: *math joke* y esta búsqueda produce mejores resultados. Por el tipo de ejemplo seguramente encontraremos una gran cantidad de artículos de *usenet*, después de ésto estamos bastante cerca, pero sería preferible estrechar algo más el criterio de búsqueda.

Para hacer esto, *veronica* nos permite especificar ciertos calificadores, a saber: *and*, *or* y *not*, además podemos agrupar las palabras entre paréntesis. Cuando se usa *and*, se indica a *veronica* que busque elementos que contengan más de una palabra. Por omisión, *veronica* considera que se usa *and* siempre que se especifica más de una palabra. Por ejemplo, las dos siguientes listas de búsqueda son equivalentes:



*math jokes*  
*math and jokes*

Se puede usar *or* para decirle a *veronica* que busque elementos que contengan cualquiera de las palabras especificadas. Por ejemplo, para encontrar todos los elementos que contengan la palabra *jokes* o la palabra *humor* (o ambas), se busca:

*jokes or humor*

Se pueden usar paréntesis para indicar que una combinación particular debe ser tratada como una sola palabra. Por ejemplo, para encontrar todos los elementos que contengan la palabra *math* y al menos una de las palabras *jokes* o *humor*, se busca:

*math and (jokes or humor)*

Como hemos mencionado, *veronica* asumirá que estamos usando *and* por default, por lo que la última búsqueda es equivalente a:

*math (jokes or humor)*

Finalmente, se puede usar *not* para indicar que el elemento no debe contener una determinada palabra. Por ejemplo, para encontrar todos los elementos que contengan las palabras *math* y *jokes*, pero no la palabra *wanted*, se debería buscar:

*math jokes not wanted*

Habrán veces que queramos buscar (o no buscar) palabras que sean similares a alguna otra, por ejemplo, podemos querer buscar elementos que contengan *mathematics* además de *math*, o *jokes* además de *joke*. Para hacer esto, se usa el carácter \* (asterisco) al final de una palabra que representa cualquier número de caracteres extra. Por ejemplo, para encontrar todos los elementos que contengan cualquier palabra que comience con *math* y cualquier palabra que comience con *joke*, se debería buscar:

*math \* joke \**



Para encontrar todos los elementos que contengan las mismas palabras, pero que no contengan cualquier otra palabra que comience por *want* o *need*, se busca:

*math \* joke \* not (want \* or need\*)*

Lo último que mencionaremos sobre búsquedas es recordar que hay dos opciones según el tipo de búsqueda con *veronica* que queramos usar, a modo de ejemplo aquí están dos líneas de un menú *veronica* convencional:

1. *Search gopherspace using veronica at NYSERNet <?>*
3. *Search gopherspace for GOPHER DIRECTORIES (NYSERNet) <?>*

La búsqueda normal (número 1 en nuestro ejemplo) busca todo tipo de elementos de menús, mientras que la búsqueda de directorios (número 3) sólo busca elementos que son, ellos mismos menús. Si buscamos algo que sabemos que es un menú, debemos usar la búsqueda de directorios, y no sólo la búsqueda será más rápida, sino que los resultados tendrán menos elementos irrelevantes.

Para aprender más sobre *veronica*, hay dos sitios donde asistir, ambos accesibles desde la mayoría de los menús de *veronica*. Primero, se puede leer el artículo "How to compose *veronica* queries". Este explica todos los detalles de como decirle a *veronica* lo que queremos. Segundo, se puede leer la lista de preguntas más frecuentes sobre *veronica*. Esta contiene muchas preguntas y respuestas sobre *veronica*.

Algo de historia y del funcionamiento de *veronica* que obtuvimos en alguno de esos sitios es lo siguiente : La primera versión de *veronica* se desarrolló en Noviembre de 1992. El trabajo fue hecho por Steven Foster y Fred Barrie de la Universidad de Nevada en Reno, Departamento de Sistemas de Computación. Foster y Barrie eran usuarios de *gopher* que comprendieron que el potencial de *gopher* nunca podría ser alcanzado a menos que hubiera una manera fácil de encontrar elementos específicos en el *gopherspacio*. Ellos veían lo frustrante e improductivo que era perseguir elementos específicos cuando no se sabía donde buscar.



Inspirados en *archie*, que había sido desarrollado para proporcionar búsquedas en bases de datos de servidores *ftp anónimos* en **INTERNET**,. Desarrollaron lo que ellos llamaron un *servidor veronica* que proporcionase una función similar para *gopher*.

De forma regular (cada una o dos semanas), su servidor *veronica* contactaba con cada servidor *gopher* conocido y pedía una copia de todos los menús. Estos menús eran almacenados en una base de datos. En cualquier momento, un usuario podía utilizar un programa de búsqueda *veronica* para buscar elementos en la base de datos.

No pasó mucho tiempo hasta que *veronica* formó parte integral de la cultura *gopher*. Hoy, hay un gran número de servidores *veronica* a lo largo de **INTERNET**, y los nuevos trabajos sobre *veronica* están coordinados con el desarrollo principal de *gopher*.

Por supuesto, *veronica* no es perfecto, es demasiado fácil que aparezcan un gran número de elementos irrelevantes o incluso que no aparezca ninguno. Algo de esto tiene que ver con como funciona *veronica*; algo de esto refleja la naturaleza del *gopherespacio* en si mismo. Sobre todo, cada servidor *gopher* es mantenido localmente y no hay estándares globales de como nombrar u organizar los elementos de un menú.

### ⌘ *jughead*

*jughead* es una herramienta similar a *veronica*, excepto en que sólo busca en un área limitada del *gopherespacio*. *jughead* fue desarrollado por Rhett (Jonzy) Jones en el Centro de Cálculo de la Universidad de Utah y fue liberado por primera vez el 25 de Marzo de 1993.

La razón de que *jughead* sea tan importante es que hay veces en las que sólo queremos buscar en una área pequeña del *gopherespacio*. Por ejemplo, un usuario de una universidad puede querer sólo buscar en los servidores *gopher* de esa universidad. Similarmente, si se usa un servicio de **INTERNET** de acceso público que mantiene una gran servidor *gopher*, sería cómodo poder buscar sólo en los elementos de los menús de ese servidor.



Lo mismo se puede decir de los usuarios externos de *gopher*. Si están interesados en encontrar algo, digamos, en una universidad en particular, les gustaría poder conectarse a uno de sus servidores *gopher* y buscar en todo el *gopherespacio* de la universidad. Cuando un administrador de sistemas decide que sus usuarios deberían beneficiarse de este tipo de búsquedas limitadas en el *gopherespacio*, configura un *servidor jughead*. El papel del *servidor jughead* es mantener una base de datos con todos los elementos de los menús de la parte del *gopherespacio* especificado por el administrador del sistema. En cualquier momento, un usuario puede usar *Jughead* para buscar lo que quiera dentro de esa área limitada.

Por ejemplo, digamos que alguien está interesado en libros y bibliotecas de una universidad en particular. Podría pensar en utilizar *veronica* para buscar *libro\* biblio\**. Sin embargo, hay tres problemas con esta solución. Primero, encontrará elementos de menús de todo el *gopherespacio*, que pueden ser demasiados. Segundo, *veronica* tiene un límite incorporado de cuántos elementos puede encontrar en una única búsqueda y ese límite puede ser alcanzado antes de encontrar algo en la universidad en particular. Y tercero, *veronica* puede tardar mucho en realizar esta búsqueda tan general.

Sin embargo, si la persona usa *jughead*, puede confinar su búsqueda a la universidad, además, la búsqueda será mucho más rápida que con *veronica* y obtendrá resultados completos. La única limitación es que alguien ha tenido que configurar ya un *servidor Jughead* para esa parte del *gopherespacio*.

Para acceder a *jughead* lo hacemos de la misma forma que lo hacemos con *veronica*, seleccionando un elemento de un menú de *gopher*. La diferencia es que, mientras que sólo hay relativamente unos pocos servidores *veronica* en el mundo, hay muchos servidores *Jughead*, cada uno ofreciendo un servicio en una parte del *gopherespacio*. Aquí hay algunos elementos típicos de menús de *Jughead*. Y para realizar una búsqueda con *jughead* es como hacerlo con *veronica*. Se introducen una o más palabras y *jughead* las busca, el resultado es devuelto como un menú de *gopher*, así que se puede seleccionar en él de la forma normal. Cuando se compone una búsqueda con *jughead*, se pueden usar las palabras especiales *and*, *or*, y *not*. Si se quiere obtener ayuda, todo lo que hay que hacer es buscar "?" (signo de interrogación) con *jughead*. Obtendremos un elemento de ayuda sobre *Jughead*, que podremos visualizar.





# INTERNET

---



## ***gopher***

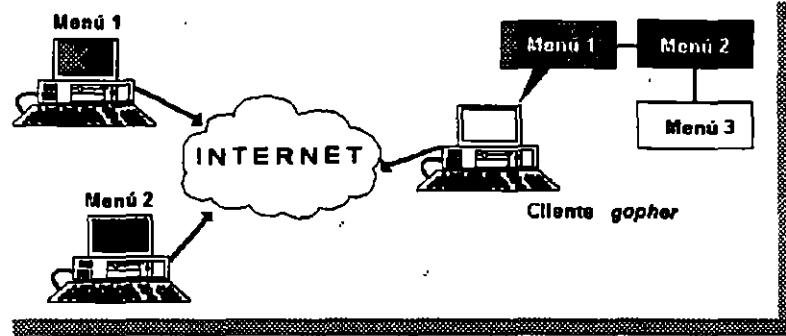
- ↳ Permite acceder los recursos de Internet a través de menús.
- ↳ Los recursos enumerados en un menú pueden estar en cualquier parte de Internet.
- ↳ Arquitectura cliente/servidor.
- ↳ La mayoría de los servidores *gopher* son públicos.
- ↳ Se define el término *gopherespacio*.

Notas:

# INTERNET



## *gopher*: Búsqueda por menúés



Notas:



# INTERNET

---



## Servicios basados en *gopher*

### ☐ *veronica*:

- ↳ Busca menús que contengan una palabra determinada.

### ☐ *jughead* :

- ↳ Similar a *veronica* , pero busca en un área moderada del *gopherespacio*.

Notas:

# CURSO: **INTERNET**

## 6.- World Wide Web



Octubre de 1995.

## 6.- World Wide Web

### INTRODUCCION

*World Wide Web*, *WWW*, *W3*, o simplemente, *Web*, es un ambicioso proyecto cuyo objetivo consiste en ofrecer una interfaz simple y consistente a la inmensidad de recursos que proporciona **INTERNET**. Para entender el *Web*, necesitamos empezar por la idea de hipertexto.

*Hipertexto* son datos que contienen *enlaces* a otros datos, un ejemplo simple de hipertexto es una enciclopedia, digamos que estamos leyendo el término "Arboles". Al final del artículo, veremos una referencia que dice, "Para más información, véase Plantas". Esta última línea es un enlace desde el artículo "Arboles" hacia el artículo "Plantas".

Por supuesto, este es un ejemplo simple. El *Web* está basado en hipertexto que es mucho más complejo. En particular, puede haber enlaces en cualquier lugar dentro de un documento, no sólo al final.

Aquí hay un ejemplo imaginario, supongamos que estamos usando el *Web* para leer un artículo de hipertexto sobre árboles, cada vez que se menciona el nombre de un nuevo árbol hay un enlace y cada enlace está marcado de alguna forma que lo destaca, por decir algo, una palabra que tiene un enlace puede estar resaltada o subrayada, o puede estar identificada por un número. Si seguimos el enlace, saltamos a un artículo que trata sobre el tipo de árbol en particular, dentro del artículo principal, hay también enlaces a otros temas relacionados tales como "bosques húmedos" o "madera". Estos enlaces conducen a artículos completos, también encontraremos enlaces a términos técnicos tales como "caduco" o "conífera". Cuando seguimos uno de estos enlaces, encontraremos una definición.

En el lenguaje del *Web*, un *documento* de hipertexto es algo que contiene datos y posiblemente, enlaces a otros documentos. El programa que se usa para leer documentos de hipertexto se llama un *visualizador* o *browser*. Cuando seguimos un enlace, decimos que estamos *navegando* por el *Web*.



En teoría, no hay ninguna razón para limitarse a los datos textuales, por ejemplo, cuando seguimos un enlace en el documento sobre árboles, podríamos encontrar un dibujo del tipo de árbol especificado, o podríamos encontrar un video de una vista aérea de un bosque. Quizás podríamos incluso encontrar sonido (digamos, de un árbol cayendo).

En el mundo del hipertexto, usamos la palabra *hipermedia* para referirnos a documentos que pueden contener una gran variedad de tipos de datos y no sólo texto, en la práctica el *Web* contiene algo de hipermedia, pero la mayoría de lo que veremos es hipertexto normal, es decir material textual que se puede visualizar en una pantalla.

Lo que hace al *Web* tan potente es que un enlace puede ir a cualquier tipo de recurso de **INTERNET**: un archivo de texto, una sesión *telnet*, un *gopher*, un grupo de discusión de *Usenet*, etc. El trabajo del visualizador es actuar como una ventana a **INTERNET** siguiendo cualquier enlace que deseemos, y accediendo a cada documento usando el método apropiado.

Por ejemplo, si se sigue un enlace a un archivo de texto, el visualizador lo traerá y lo mostrará, pantalla a pantalla. Si se sigue un enlace a una sesión *telnet*, el visualizador iniciará la sesión sin necesidad de saber nada técnico como la dirección o el número de puerto. Si se sigue un enlace que conduce a un grupo de discusión de *Usenet*, el visualizador presentará los artículos usando un formato simple de hipertexto.

Por tanto, podemos caracterizar el *Web* con un poco más de precisión. El *Web* es un intento de organizar toda la información de **INTERNET** (más cualquier información local que se desee añadir) como un conjunto de documentos de hipertexto. Aunque este sueño pueda parecer poco realista, el *Web* permite acceder a todo tipo de recursos de **INTERNET**, mediante el uso de un simple visualizador para poder "leer" el documento apropiado.

El *Web* fué desarrollado originalmente en Suiza, en el centro de investigación CERN. La idea fue crear una forma para los físicos del CERN de compartir su trabajo y usar la información de la comunidad. Muy pronto, la idea del *Web* se expandió y fue adoptada dentro de **INTERNET** como un mecanismo general para poder acceder a información y servicios.



En teoría, la idea de un sistema de hipertexto a lo largo de **INTERNET** parece ofrecer el "santo grial" de la recuperación de información: un método simple y fácil de encontrar y usar cualquier tipo de datos que existan. Sin embargo, en la práctica, encontraremos que la utilidad del *Web* está generalmente limitada a dos funciones principales: leer artículos de hipertexto y acceder a recursos de información.

La idea de hipertexto no es nueva. Quizá en alguna ocasión ya hayamos utilizado este sistema, por ejemplo la ayuda de cualquier aplicación de Windows nos marca enlaces a otras palabras, o el programa Hypercard en un Macintosh, o el InfoExplorer en un IBM RS/6000. El hipertexto puede ser una gran manera de leer información, pero hay varios problemas inevitables. Primero, lo que hace al hipertexto más valioso que el texto normal es tener todos los enlaces, el problema es que poner enlaces en un hipertexto es un proceso que consume tiempo. Requiere tanto esfuerzo, que todavía no hay mucho hipertexto disponible. Hay algún trabajo hecho para intentar automatizar el proceso: hacer que un programa lea texto normal e intente calcular donde poner enlaces, sin embargo, no hay realmente un sustituto para el criterio humano.

El segundo problema consiste en que el valor añadido al texto normal por los enlaces de hipertexto depende de lo útiles que sean los enlaces, se supone que cada enlace representa un salto mental que una persona querría hacer cuando lee el texto. Desafortunadamente, muchos tipos de textos no se prestan para hacer este tipo de saltos. Además, incluso aunque los saltos sean útiles, no siempre es obvio a donde deben ir los enlaces, lo que personalmente se obtiene de un documento de hipertexto depende mucho de lo cerca que esté nuestro pensamiento del de la persona que puso los enlaces en el documento.

A pesar de eso, hay algunos tipos de información para los cuales es más o menos obvio hacia donde deben ir los enlaces, por ejemplo, un documento erudito debería contener un enlace a cada nota a pie de página. Cuando usemos el *Web* para leer tales documentos, encontraremos que el hipertexto puede ahorrarnos gran cantidad de tiempo. No sólo se puede saltar a un tema relacionado cuando surja la necesidad, sino que también se pueden pasar por alto varios detalles ignorando los enlaces que nos parezcan aburridos o triviales.





El *Web* no es perfecto. Habrá veces en que será preferible usar una herramienta diseñada específicamente para un tipo de datos en particular. Por ejemplo, aunque se pueda usar el *Web* para leer los artículos de *Usenet*, se leen mejor usando un lector de noticias normal. A pesar de eso, el *Web* es la herramienta más flexible que hay para explorar **INTERNET**.

## 🔗 BROWSERS O VISUALIZADORES

Igual que muchos otros servicios de **INTERNET**, el *Web* utiliza el sistema cliente/servidor, de tal forma que usamos un programa cliente llamado browser o visualizador que actúa como nuestra ventana al *Web*. Desde el punto de vista del *Web*, todo en el universo consiste en documentos o enlaces. Por tanto, el trabajo del visualizador es leer documentos y seguir los enlaces que seleccionemos. Un visualizador sabe como acceder a cada servicio y a cada recurso de **INTERNET**. Puede realizar una búsqueda *wais*, leer artículos de *Usenet*, acceder a los elementos de *gopher*, iniciar sesiones *telnet*, etc. Lo más importante es que un visualizador también sabe como conectarse a un *servidor WWW* que ofrece documentos públicos de hipertexto.

Hay docenas de servidores *WWW* en **INTERNET** y la mayoría de ellos están especializados en un área en particular, por ejemplo, el servidor *WWW* de la Cornell University Law School contiene una gran cantidad de documentos que tratan sobre las leyes de E.U. En general, encontraremos dos tipos de documentos. Texto, que se puede leer, e *índices*, en los que podemos buscar. Cuando el visualizador encuentra que un enlace apunta a un documento textual, recupera el documento y lo muestra, pantalla a pantalla. Mientras que cuando el visualizador encuentra un índice, muestra una descripción e invita a introducir una o más palabras clave. El visualizador entonces explora el índice y devuelve los datos que mejor encajan con las palabras clave. Una búsqueda con el *Web* es muy parecida a una búsqueda *wais*, y de hecho, muchos índices son realmente fuentes de *wais*.

Sólo hay tres habilidades básicas necesarias para usar el *Web*. Primera, se necesita ser capaz de controlar la visualización de texto, segunda, hay que saber como decirle al visualizador que siga un enlace, y tercera, hay que ser capaz de especificar como se quiere buscar en un índice.



La forma exacta de como funcionan estas tres operaciones depende de qué visualizador se utilice, si el visualizador usa una interfaz gráfica de usuario con un ratón, los enlaces estarán resaltados y se podrá seleccionar uno simplemente pulsando sobre él, con un visualizador basado en caracteres, cada enlace tendrá asignado un número (estilo *gopher*) y para seleccionar un enlace hay que introducir el número. A pesar de eso, no importa que tipo de visualizador se tenga, usar el *Web* es fácil e intuitivo.

Como sucede con la mayoría de los servicios de **INTERNET**, la experiencia trabajando con el *Web* depende mucho del programa cliente (visualizador) que se utilice. La mejor forma de acceder al *Web* es usar un visualizador que esté diseñado para obtener todas las ventajas de las características especiales del sistema. Por ejemplo, si utilizamos *X-Window* lo recomendable es ejecutar un visualizador *X-Window* que hará uso del ratón y de la interfaz gráfica de usuario.

Existe un gran número de visualizadores que se ejecutan sobre diferentes plataformas, *X-Window*, PC con Microsoft Windows, Macintosh, VMS, varios tipos de Unix, etc. Pero además existe también la opción de ejecutar un cliente *WWW* remoto, haciendo *telnet* a uno los hosts listados a continuación:

**Visualizadores de *Web* de uso público a los que se puede hacer *telnet*.**

Localización	Dirección INTERNET	Dirección IP
Finlandia	info.funet.fi	128.214.6.102
Hungría	fserv.kfki.hu	148.6.0.3
Israel	vms.huji.ac.il	128.139.4.3
Eslovaquia	sun.uakom.cs	192.108.131.11
Suiza	info.cern.ch	128.141.201.74
USA: Kansas	ukanaix.cc.ukans.cdu	129.237.1.30
USA: Nueva Jersey	www.njit.edu	128.235.163.2
USA: Nueva York	fatty.lawcornell.edu	132.236.108.5

Si se pide un identificador de usuario, hay que usar **www**.

Cada uno de estos hosts ofrece un visualizador de acceso público que puede usar cualquiera. En algunos hosts, el visualizador se ejecutará automáticamente, en otros hay que conectarse como *www*.



Todos estos hosts ofrecen el tipo de visualizador que puede usar remotamente todo el mundo en **INTERNET**. Esto significa que el visualizador debe asumir que estamos usando un terminal ordinario sin características especiales. Hay dos tipos de visualizadores que se usan para este propósito, y cuando se hace *telnet* a uno de estos hosts públicos, se usará uno de estos visualizadores.

Un visualizador en *modo línea* es el tipo más simple, ya que muestra la salida escribiendo una línea tras otra en la terminal, es decir que se ve como si todo apareciese en la última línea de la pantalla y se desplazase hacia arriba. Un visualizador en modo línea no se preocupa del tipo de terminal que se usa, ya que todo lo que hace es enviar una línea de texto tras otra.

Un visualizador en *modo pantalla* utiliza la pantalla completa, al igual que el visualizador en modo línea, un visualizador en modo pantalla sólo usa caracteres, con la diferencia que puede escribir esos caracteres en cualquier parte de la pantalla, trayendo como consecuencia que el visualizador sea más rápido, y permite utilizar una interfaz más compleja. Sin embargo, también significa que el visualizador necesita conocer el tipo de pantalla que se está utilizando.

Cuando conectamos con un sistema que utiliza un visualizador en modo pantalla, se nos pedirá que especifiquemos el tipo de terminal. Ya que la mayoría de los sistemas emulan VT-100 para las sesiones *telnet* remotas, éste será el que utilice por default.

Cuando manejamos un visualizador independientemente de su ubicación local o remota, el formato de la dirección desde la que estamos accedando al hipertexto es diferente al formato tradicional de correo electrónico o al sistema de nomenclatura de dominios, esta información se llama un *Localizador Uniforme de Recursos* o *URL*.



Dentro de **INTERNET**, cada recurso puede ser descrito por un *URL*. De forma rápida la primera parte de un *URL* especifica qué tipo de recurso se está describiendo, por ejemplo *http* indica que estamos viendo un documento de hipertexto. (El nombre *http* viene de Hypertext Transport Protocol - Protocolo de Transporte de Hipertexto, el protocolo que usa el *Web* para mover los datos de un lugar a otro). La segunda parte contiene el nombre de la computadora (en base al sistema de nomenclatura de dominios) y la tercera parte contiene una ruta de acceso dentro de esa computadora que nos permite llegar hasta el documento que estamos visualizando.

A continuación se presenta un resumen de las principales instrucciones que podemos ejecutar con una sesión de *WWW* en modo línea.

### **WWW:Comandos en modo línea**

<b>Instrucción</b>	<b>Nombre Completo</b>	<b>Descripción</b>
number	number	seguir el enlace especificado
b	back	saltar al documento previo de la lista
bo	bottom	saltar al final del documento actual
f <i>palabras</i>	find <i>palabras</i>	usar palabras clave especificadas para buscar en un índice
g <i>URL</i>	go <i>URL</i>	saltar al documento o recurso especificado
h	help	mostrar un sumario de órdenes e información técnica
ho	home	saltar al documento de inicio
l	list	mostrar los enlaces dentro del documento actual
m	manual	saltar al documento de referencia sobre el visualizador
n		saltar al siguiente enlace dentro del último documento
p	previous	saltar al enlace anterior dentro del último documento
quit	quit	salir del <i>Web</i>
r	recall	mostrar la lista de documento visitados previamente
r <i>número</i>	recall <i>número</i>	saltar al documento especificado de la lista
RETURN	-	mostrar la siguiente pantalla del documento actual
t	top	saltar al principio del documento actual
u	up	mostrar la pantalla anterior del documento actual



Hay programas visualizadores para una gran variedad de sistemas diferentes disponibles vía *ftp anónimo* en *info.cern.ch*. Para más información, hay que buscar en el directorio llamado */pub/www* un archivo llamado *readme.txt*.

Si tiene un sistema X-Window, de las mejores elecciones es ejecutar el *Mosaic*. Mosaic ha sido escrito por Marc Andreessen del National Center for Supercomputing Applications (NCSA) en la Universidad de Illinois en Urbana. Mosaic tiene un gran número de características avanzadas, tales como la inclusión gráficos y el lanzamiento de aplicaciones multimedia. También usa colores para ayudar a seguirle la pista a los enlaces. Por ejemplo, Mosaic recuerda cada documento que se ha leído, por lo que cuando un enlace apunta a algo que ya hemos visto, Mosaic pintará ese enlace usando un color especial. Esto ayuda a evitar perder tiempo en documentos que ya se han visto, también hace más fácil seguir el camino recorrido a través de territorio desconocido.

Otra característica interesante es que Mosaic permite añadir notas personales dentro de cualquier documento. Estas notas aparecerán cada vez que leamos el documento. (Por supuesto, las notas son privadas, sólo las podemos ver nosotros).

El aspecto de Mosaic es el de una aplicación Motif, aunque no es necesario tener Motif en la computadora. Instalar Mosaic es fácil, viene preparado para ejecutarlo directamente. Todo lo que se necesita hacer es cargar el programa, descomprimirlo (usando la orden *uncompress*) y hacerlo ejecutable (usando la orden *chmod*). No hay nada que configurar.

Mosaic está disponible vía *ftp anónimo* en *ftp.ncsa.uiuc.edu*. El nombre real del programa es *xmosaic*. Hay que ir al directorio */Web/xmosaic-binaries* y buscar la versión de nuestro sistema.

Además existe un grupo de discusión de *Usenet* al que nos podemos subscribir llamado *comp.infosystems.www*.

El visualizador más comúnmente utilizado para las PC's y Mac's es el llamado *Netscape* el cual funciona de manera similar a Mosaic.



# INTERNET



---

## World Wide Web

☐ Conocido también como:

- ↳ WWW
- ↳ W3
- ↳ Web

☐ Objetivo:

Ofrecer una interface simple y consistente a la  
inmensidad de recursos que proporciona Internet.

Notas:

# INTERNET

---



## World Wide Web

- ↪ Está basado en el Hipertexto.
- ↪ Un documento de Hipertexto contiene datos y Posiblemente enlaces a otros documentos.
- ↪ Los datos pueden ser textos, imágenes, sonidos y animaciones.
- ↪ Surge la hipermedia.

Visualizador o browser.- Programa para leer un documento de Hipertexto.

Notas:

# INTERNET

---



## World Wide Web

- ↳ Intento por organizar toda la información de Internet.
- ↳ El Web no es perfecto.
- ↳ Existen algunas limitaciones.
- ↳ Arquitectura cliente/servidor.

Notas:



# INTERNET

---



## World Wide Web

### Visualizadores:

Modo línea.

X-Window.

Mosaic.

Windows y Mac.

Netscape ( PC's, Mac's ).

**Notas:**

# INTERNET



---

## World Wide Web

**Formato de entrada**

**`http: // www. njit.edu`**

**Notas:**

# INTERNET



## World Wide Wed

Overview of the Web (23/27)

### GENERAL OVERVIEW

There is no "top" to the World-Wide Web. You can look at it from many points of view... If you have no other bias, here are some places to start:

- |               |  |
|---------------|--|
| by Subject[1] | A classification by subject of interest. Incomplete but easiest to use.  |
| by Type[2]    | Looking by type of service (access protocol, etc) may allow you to find things if you know what you are looking for. |
| About WWW[3]  | About the World-Wide Web global information sharing project.   |

#### Starting somewhere else

To use a different default page, perhaps one representing your field of interest, see "customizing your home page"[4].

1-6, Up, <RETURN> for more, Quit, or Help:

Notas:

# CURSO: **INTERNET**

## 7.- OTROS SERVICIOS DE INTERNET



Octubre de 1995.

## 7.- OTROS SERVICIOS DE INTERNET

### ☒ TIPOS DE ARCHIVOS

#### ☞ INTRODUCCION

Para comprimir o empaquetar archivos existen diversos programas de utilería, para el caso de sistemas Unix, los más comunes son: *compress*, *uncompress*, *tar*, *uuencode*, *uudecode*, *btoa* y *shar*, si se utiliza una PC o una Macintosh, existen versiones de dominio público de estos programas disponibles vía *ftp anónimo*.

Existen dos maneras sencillas de encontrar estos programas. Primero, puede comprobarse el archivo de referencia de utilerías que es mantenido como un servicio público por David Lemson de la Universidad de Illinois. Para obtener este documento, debe hacerse *ftp* a *ftp.cs.uiuc.edu* y buscar en el directorio */doc/pcnet* el archivo *compression*.

La segunda forma de localizar un programa de utilerías de archivos es utilizar *archie*, el método más fácil es utilizar una búsqueda *regex* y buscar el nombre del programa que se quiere obtener. Para ganar tiempo, puede especificarse un patrón de búsqueda que empiece con un caracter "*^*" (circunflejo). Esto significa que el patrón debe encontrarse al principio de una palabra, por ejemplo, puede buscarse *^tar* esto localizaría *tar.exe* pero no *game.tar.Z*.

Veamos tres ejemplos que buscan el programa *tar*. Primero utilizando un cliente *archie*:

```
archie -m25 -r ^tar more
```

Segundo, utilizando *archie* por correo, enviando un mensaje que contiene las órdenes similares a:

```
set mailto pedro.huerta@industry.net
set maxhits 25
set sortby time
set search regex
find ^tar
quit
```



Tercero, estableciendo una conexión *telnet* con el usuario *archie* en un servidor *archie* y tecleando las órdenes:

```
set maxhits 25
set pager
set sortby time
set search regex
find ^tar
```

#### TIPOS DE ARCHIVOS UTILIZADOS CON *ftp anónimo*

Las personas que configuran los servidores de *ftp anónimo* tienen que considerar dos problemas muy importantes sobre el almacenamiento de los archivos, para empezar, muchos archivos son muy grandes y requieren una gran cantidad de espacio en disco, y por ende requieren mucho más tiempo de transmisión que los archivos pequeños. Segundo, muchos de los elementos ofrecidos al público son paquetes de software y no simples archivos. Una simple pieza de software, por ejemplo, puede constar de varios programas separados y archivos de documentación. En estos casos, el paquete necesita ser fácil de almacenar y transmitir.

Por estas razones, los archivos *ftp anónimo* se almacenan en dos clases de formatos especiales. Primero, los archivos muy grandes se *comprimen* en un formato que requiere menos espacio, pero cuando obtenemos un archivo de estos, debemos utilizar un programa para *descomprimir* el archivo antes de poder utilizarlo. Segundo, un grupo de archivos puede recopilarse dentro de un solo archivo, el cual puede almacenarse y transmitirse como un solo archivo.

Estos archivos pueden transmitirse de igual forma que un archivo normal, y una vez que se ha copiado el archivo a la computadora local, debe utilizarse un programa para recuperar la colección de archivos original. Cuando se hace esto, se dice que se está *desempaquetando* un archivo o que se están *extrayendo* los archivos.



Existen una serie de programas que se utilizan para comprimir y recopilar archivos, dentro de las máquinas Unix el más comúnmente utilizado para compresión se denomina *compress*. Cuando se utiliza este programa, crea un archivo nuevo que tiene el mismo nombre que el archivo que se ha comprimido seguido de los caracteres ".Z", por ejemplo, cuando se comprime el archivo *sei*, se genera un archivo nuevo (más pequeño) llamado *sei.Z*, es por ello que siempre que encontremos un archivo cuyo nombre termina en ".Z", sabremos que se trata de un archivo comprimido que deberá descomprimirse antes de poder ser utilizado. Para ello, debe utilizarse el programa *uncompress*.

Cuando es necesario recopilar varios archivos dentro de uno solo, el formato más utilizado es *tar*. Y cuando lo utilizamos, creamos un archivo nuevo cuyo nombre termina en ".tar", por ejemplo, podemos recopilar un grupo de archivos que contienen todos los programas y documentación de paquete de software denominado *game* en un archivo llamado *game.tar*. En general este tipo de archivos se denominan *archivos tar*. Por lo que cuando encontremos archivos cuyos nombres terminan en tar, sabremos que se trata de un archivo que debe desempaquetarse antes de utilizar los archivos. Para lo cual debemos utilizar el programa *tar* en la computadora local.

Al recopilar archivos separados dentro de un solo archivo, no estamos comprimiendo los datos de estos archivos, luego es común que después de crear un archivo *tar*, éste sea comprimido, lo cual nos da como resultado archivos con extensiones ".tar.Z", por ejemplo *game.tar.Z*

Esto significa que un grupo de archivos se ha recopilado dentro de un único archivo que posteriormente fue comprimido. Cuando se obtiene uno de estos archivos, primero debe utilizarse *uncompress* para descomprimir el archivo y después hay que utilizar *tar* para recuperar los archivos originales. Si no se recuerda la orden de ejecución de estas operaciones, simplemente debe mirarse el nombre del archivo y proceder de derecha a izquierda.

En una PC basada en DOS, el nombre de un archivo consta de un "nombre\_de\_archivo" y una "extensión" opcional, el nombre\_de\_archivo no puede contener más de ocho caracteres, mientras que la extensión no puede tener más de tres caracteres, por eso, los archivos *tar* comprimidos usados en DOS utilizan la extensión ".taz." Por ejemplo, podemos encontrar un archivo llamado *game.taz*, y para obtenerlo debemos copiarlo a nuestra PC y después descomprimirlo y desempaquetarlo.



Dentro de las PC's también es muy conocido el programa *pkzip*, el cual empaqueta y comprime al mismo tiempo, y para descomprimir y desempaquetar estos archivos, hay que utilizar un programa denominado *pkunzip*. (Las iniciales *pk* corresponden a Phil Katz, el creador del programa). El programa *pkzip* genera archivos con la extensión "*zip*", por ejemplo *sei.zip*. Estos archivos suelen denominarse archivos *zip*.

## TIPOS DE ARCHIVOS UTILIZADOS EN *Usenet* Y EN CORREO ELECTRONICO

Hablando de forma general, cualquier archivo puede clasificarse como texto o binario, los archivos de texto contienen caracteres ordinarios: letras, números y signos de puntuación. Estos archivos pueden utilizarse directamente, esto es que podemos ver el contenido de un archivo de texto utilizando un programa de paginación, o editar un archivo de texto utilizando un editor de textos. Otro nombre por el que se conocen los archivos de textos es archivos ASCII, dado que los caracteres se almacenan de acuerdo con un sistema denominado código ASCII. (ASCII corresponde a American Standard Code for Information Interchange).

Los archivos binarios contienen datos que deben utilizarse por un determinado programa, analicemos un ejemplo cotidiano. Muchos usuarios utilizan el servicio *ftp anónimo* y *Usenet* para distribuir imágenes que pueden visualizarse en la pantalla de la computadora. Estas imágenes se almacenan en archivos binarios y se pueden ver mediante un programa de visualización de imágenes que reconozca el formato del archivo. Si se pretende ver un archivo de imagen con un programa de paginación diseñado para mostrar archivos de texto, veremos una cadena de código ASCII y no tendrá sentido. Y lo mismo ocurrirá con los archivos binarios que contienen sonidos que pueden escucharse en computadoras con el hardware apropiado. En este caso, será necesario un programa de sonido que reconozca el formato del archivo binario. Otros ejemplos son los procesadores de texto, hojas de cálculo y bases de datos que almacenan los datos en archivos binarios de acuerdo con un formato propio.

En el Capítulo 2, se explicó que los programas de correo electrónico sólo pueden contener texto, no datos binarios a menos que se utilice un protocolo especial denominado *MIME*: (Multipurpose INTERNET Mail Extensions), y de la misma forma los artículos de *Usenet* también se componen de texto ordinario.





Esto plantea un problema: ¿Cómo se puede enviar a otro usuario por correo un archivo binario? ¿Cómo se puede enviar un archivo binario -como una imagen- a un grupo de discusión de *Usenet*?. La solución es convertir el archivo binario en un archivo tipo texto, después de esto, puede enviarse por correo o a un grupo de *Usenet*. El usuario que reciba este archivo de texto debe convertirlo de nuevo a binario.

Dentro de Unix existen dos programas que suelen utilizarse para convertir archivos de binario a texto. El primero se denomina *uuencode*. A los archivos que crea *uuencode* se les denomina *archivos uuencode*. Para convertir un archivo *uuencode* a un archivo texto se utiliza *uudecode*. (Los nombres de estos dos programas se deben a que fueron desarrollados como parte de UUCP, un sistema que se puede utilizar para copiar archivos de una computadora Unix a otra (UUCP corresponde a las siglas de la frase "Unix to Unix copy program"). Cuando se almacenan archivos *uudecode* en un servidor *ftp anónimo*, normalmente se les asigna la extensión *.uue*, por ejemplo, *sei.uue*. A veces estos archivos se comprimen, en tal caso tendrán un nombre del tipo *sei.uue.Z*.

El otro programa que se utiliza para convertir archivos binarios a texto se llama *btoa* (binary to ASCII). A los archivos que crea *btoa* se les conoce como *archivos btoa*. No existe una extensión especial para identificar estos archivos.

El último programa que se discutirá en esta sección será *shar*. Se utiliza para crear archivos para enviar por correo o a *Usenet*. Los archivos creados por *shar* se denominan *archivos shar*. Cuando se almacenan archivos *shar* en servidores *ftp anónimo*, se les asigna la extensión *.shar*, por ejemplo, *sei.shar*. A veces, estos archivos se comprimen, y aparecerán con la extensión *sei.shar.Z*. La ventaja de utilizar un archivo *shar* es que es muy fácil de desempaquetar, el archivo *shar* es un programa ejecutable, el cual automáticamente se desempaqueta y genera los archivos originales en cuanto lo corremos. El programa *shar* utiliza el intérprete de órdenes Bourne shell y puede ejecutarse utilizando la instrucción *sh*. También existe un programa denominado *unshar* que permite desempaquetar estos archivos.

Antes de copiar un archivo binario vía *ftp* debemos preparar la transferencia ejecutando la orden *binary*, ya que de no hacerlo el programa *ftp* asumirá que vamos a transferir un archivo de texto ASCII y no se copiará correctamente.



Los archivos comprimidos *.Z*, *.tar*, *.taz* y *.zip* son siempre binarios, los archivos *.uue*, *.shar*, y *btoa* son siempre archivos de texto.

Existen además dos tipos de archivos de documentos que aparecen comúnmente en los servidores *ftp anónimo*, el primero, son los archivos de texto normales, los cuales suelen tener la extensión *".txt"*, por ejemplo, *document.txt*. Y podemos visualizarlo con cualquier editor o programa de paginación. El otro tipo común de un archivo de documento tiene la extensión *".ps"*, que son archivos PostScript, y pueden contener tanto imágenes como caracteres. Técnicamente, los archivos PostScript son archivos de texto y es posible ver su contenido con un editor de textos, sin embargo, los datos se almacenan en formato PostScript y no tendrán sentido para un usuario normal. Para obtener la salida de un archivo PostScript, o bien se imprime por una impresora PostScript, o se utiliza un programa que pueda visualizar archivos PostScript por pantalla.

Si utilizamos una PC, debemos tomar en cuenta que el programa de compresión y archivado más popular es *pkzip* y su hermano *pkunzip*. El programa *pkunzip* lee un archivo *zip* y restaura los archivos originales, por lo que es a la vez un programa de desempaquetado y descompresión.

La versión de PC del programa *pkzip* es *shareware*. Esto significa que el programa puede distribuirse y utilizarse libremente de forma gratuita, sin embargo, si el programa se utiliza asiduamente, debería enviarse una pequeña cuota a la compañía que desarrolla *pkzip* (cuyo nombre es PKWare). Entonces se pasa a ser un usuario registrado y se obtienen ciertas ventajas, como recibir un manual impreso.

## ☞ "BAJAR" ARCHIVOS A NUESTRA COMPUTADORA DESDE INTERNET

Si tenemos el software adecuado, cargar y descargar archivos entre una computadora personal e **INTERNET** es fácil, pero si no tenemos el software adecuado o si no funciona adecuadamente copiar archivos puede ser imposible. *Cargar* archivos se refiere a copiar archivos desde una computadora remota a la computadora local, mientras que *descargar* archivos significa copiar archivos de la computadora local a un host remoto.



Antes de empezar, recordemos que hay dos formas de utilizar una PC (o Macintosh) para acceder a **INTERNET**, en el caso más simple, la PC estará conectada directamente a **INTERNET**. Sin embargo, si trabajamos en casa, probablemente utilizemos la PC para ejecutar un programa de comunicaciones, el cual utiliza un modem para llamar a un número de teléfono y conectarnos con un host remoto de **INTERNET**. El programa de comunicaciones nos permite emular un terminal y podemos iniciar entonces una sesión de trabajo con un identificador de usuario y una palabra clave.

Si la PC está conectada directamente a **INTERNET**, los archivos pueden cargarse vía *ftp anónimo*. Sin embargo, si utilizamos un programa de comunicaciones para conectarnos a un host remoto, los archivos *ftp* se enviarán a dicho host. Después, debemos copiarlo del host a nuestra PC y es de esto de lo que vamos a tratar en esta sección.

Para copiar archivos de una computadora a otra, se utiliza lo que se denomina un *protocolo de transferencia de archivos*. En el mundo de las PC, el protocolo de transferencia más común se denomina *Zmodem*. De este modo, para descargar y cargar archivos, necesitamos tener el software *Zmodem* en la computadora local y en la computadora remota. Todos los programas de comunicaciones de PC vienen con funcionalidad *Zmodem*. Es necesario asegurarse de que existe este protocolo en el host.

En los sistemas Unix, el programa *Zmodem* más conocido es *sz* (que significa "send using *Zmodem*") y su compañero *rz* ("receive using *Zmodem*"). Puede visualizarse la documentación de estos programas con el comando *man*.

Para cargar un archivo desde un host remoto a una PC hay que escribir *sz*, seguido de las opciones *-a* o *-b* y a continuación el nombre del archivo. Por ejemplo:

```
sz -a document
sz -b picture.gif
```



Si se está cargando un archivo de texto (ASCII), se utiliza la opción *-a*. Mientras que si se trata de un archivo binario se utiliza *-b*. Si no se está seguro del tipo de archivo, deberá utilizarse *-b*. Tan pronto como se teclea la orden *sz*, el programa de comunicaciones puede detectar que se ha iniciado una transferencia *Zmodem* y comenzará la transferencia del archivo de forma automática, o bien el programa de comunicaciones nos pedirá que habilitemos la opción de *Zmodem* para que comience una transferencia. La forma de hacer esto depende del programa de comunicaciones, por lo que debe leerse la documentación que viene con el programa.

Para descargar un archivo desde la PC a un host remoto, deberá teclearse una de las siguientes órdenes (en el host):

```
rz -a  
rz -b
```

Después de teclear esta orden, aparecerá un mensaje que nos indica que el host está listo y hay que indicarle a nuestra PC que vamos a comenzar la transmisión por *Zmodem*.

 **wais**

## INTRODUCCION

*wais* es una herramienta que permite buscar grandes cantidades de información de forma rápida y completa. El nombre *wais* viene de "Wide Area Information Service" (Servicio de Información de Area Extensa). Como servicio de **INTERNET**, *wais* es único ya que se desarrolló a partir de un proyecto que comenzaron tres empresas comerciales: Apple, Thinking Machines, y Dow Jones.

La idea original era muy ambiciosa. En un mundo con demasiada información, una computadora podría seguirle la pista a una gran cantidad de datos, separarlos, y presentar sólo la información relevante para unas necesidades concretas. La idea era crear un programa que actuase como un bibliotecario personal, ahorrándonos tiempo y haciendo posible que accedáremos a información que nunca hubiésemos sabido que existía.



La idea original de *wais* era desarrollar un sistema generalizado de recuperación de información que fuera capaz de acceder a colecciones de datos de todo el mundo. Algunos de los datos estarían disponibles gratuitamente a través de **INTERNET**, pero no pasaría mucho tiempo antes de que la gente empezase a pagar por la información. Lo que le ocurrió a *wais* era algo predecible, al menos visto con posterioridad. No ha resultado ser la utilidad de recuperación de información del futuro. Si, hay gente que utiliza sistemas automáticos tipo *wais* por los que paga dinero, pero la mayoría de nosotros aún dependemos del papel tradicional. Así, con respecto a **INTERNET**, Thinking Machines ha dejado de soportar la distribución pública de *wais*. Una versión gratuita de *wais*, llamada *Freewais*, es mantenida por una organización llamada el CNIDR (Clearinghouse for Networked Information Discovery and Retrieval).

*wais* es un servicio de **INTERNET** que puede explorar cientos de colecciones de datos. Cada una de estas colecciones de datos se llama una *fuentes*. Le decimos a *wais* que fuentes debe usar y lo que queremos que encuentre. Para ello, se especifican una o más palabras, llamadas *palabras clave*, que queremos que *wais* busque, entonces *wais* explorará el texto completo de cada elemento de las fuentes especificadas, y encontrará las que cumplan el criterio. Esto se llama una *búsqueda textual*.

Usar *wais* es razonablemente sencillo. Todo lo que se hace es seleccionar una o más fuentes y decirle a *wais* lo que tiene que buscar, él se conectará con la computadora que contenga las fuentes y le pedirá que haga la búsqueda, mostrándonos como salida una lista de artículos o citas.

*wais* visualizará la lista, mostrando los elementos de más a menos *pertinente*. Lo que *wais* entiende como pertinente se basa en la frecuencia con la que aparecen las palabras clave en cada artículo. *wais* considera los artículos más pertinentes aquéllos en los que aparecen más veces las palabras clave. Una vez que *wais* muestra lo que ha encontrado, nos podemos mover a través de la lista y escoger los elementos que queramos ver, acto seguido *wais* entonces recuperará el texto real y lo mostrará en nuestra pantalla, si queremos conservar el elemento, le podemos decir a *wais* que lo guarde en un archivo o que nos lo envíe por correo.

Cuando encontramos un artículo especialmente interesante, se le puede decir a *wais* que examine todo o parte del artículo y use esas palabras en búsquedas posteriores. Esto se llama *pertinencia realimentada*.



Pese a que el servicio que *wais* proporciona puede ser extremadamente útil, existen ciertas limitaciones. Primera, *wais* sólo puede ayudarnos si tiene acceso a la información que queremos, aunque varios cientos de fuentes de información parecen muchas, no son todas las que podemos disponer en **INTERNET**. Segunda, *wais* examina virtualmente cada palabra contenida en una fuente ignorando algunos artículos como "un" o "el" pero sólo de una forma simple y sin atender al contexto.

## ☞ CLIENTES *wais*

Al igual que los demás servicios de **INTERNET**, *wais* utiliza el sistema cliente/servidor, de tal forma que necesitamos interactuar directamente con un programa *cliente wais*. Es él quien muestra información y lleva a cabo nuestras órdenes, cada fuente de información *wais* es mantenida por un programa *servidor wais*. Hay muchos servidores públicos *wais* dentro de **INTERNET**, y todo lo que se necesita para poder acceder a ellos es un cliente *wais*.

Siempre que hacemos una petición, el cliente *wais* se conecta con el servidor *wais* apropiado y le pide que realice la búsqueda, el servidor lleva a cabo la búsqueda y envía los resultados al cliente, el cual a su vez nos los muestra en la pantalla. Cuando indicamos que queremos ver un elemento en particular, el cliente de nuevo envía la petición al servidor, el servidor envía ahora el elemento que el cliente muestra pantalla a pantalla.

Hay gran variedad de clientes *wais* disponibles gratuitamente vía *ftp anónimo*. Para el caso de sistemas Unix los dos de los clientes *wais* más comunes son *swais* y *waissearch*, para el caso de X-Window es *xwais* y en general existen varios clientes *wais* para las diferentes plataformas.

Si no contamos con un cliente *wais* en nuestra computadora, podemos hacer *telnet* a una de las computadoras listadas a continuación, y conectarnos como se indica.



## CLIENTES *wais* DE ACCESO PUBLICO A LOS QUE SE PUEDE HACER *telnet*

Localización	Dirección INTERNET	Dirección IP	Conectarse como...
Finlandia	info.funet.fi	128.214.6.102	<i>wais</i>
USA: California	swais.cwis.uci.edu	128.200.15.2	<i>swais</i>
USA: Massachusetts	npsc.nsf.net	128.89.1.178	<i>wais</i>
USA: Massachusetts	quake.think.com	192.31.181.1	<i>wais</i>
USA: Carolina del N.	kudzu.cnidr.org	128.109.130.57	<i>wais</i>
USA: Carolina del N.	sunsite.unc.edu	152.2.22.81	<i>swais</i>

Cada una de estas computadoras proporciona un cliente *wais* de acceso público que cualquiera puede usar, una vez realizada la conexión el cliente se ejecutará automáticamente. Aunque hay que destacar es que los diferentes hosts con clientes *wais* de acceso público no ofrecen el mismo conjunto de fuentes, por lo que hay que probar todos los hosts para ver lo que hay disponible.

Cuando hacemos *telnet* a un cliente *wais* de acceso público, utilizamos el cliente *swais*. En el resto de la sección, asumiremos que es ese el cliente que estamos utilizando. Si se usa un cliente diferente, habrá que leer la documentación apropiada. Sin embargo, los conceptos básicos de cómo usar *wais* serán los mismos que los que discutiremos.

Para obtener más información sobre *wais*, hay dos grupos de discusión de *usenet* a los que uno se puede subscribir: *comp.infosystems.wais* y *alt.wais*, además hay una lista de preguntas más frecuentes (FAQ) sobre *wais* que puede ser útil, especialmente si se busca un cliente *wais* para una computadora. La lista FAQ se envía regularmente a *comp.infosystems.wais*. También se puede obtener vía *ftp anónimo* de *rtfm.mit.edu*. Hay que buscar en el directorio llamado */pub/usenet/news.answers/wais-faq* un archivo llamado *getting-started*.



Para ejecutar *wais*, hay que hacer *telnet* a uno de los clientes de acceso público, cuando la conexión *telnet* se ha establecido, hay que introducir el identificador de usuario *wais* o *swais*, con lo cual el sistema nos preguntará algún otro identificador de usuario como por ejemplo nuestra dirección de correo y nos preguntará después por el tipo de terminal con la que vamos a trabajar. Una vez respondidas todas estas peticiones, veremos la lista de fuentes.

La pantalla de *wais* nos indica el número total de fuentes que reconoce el servidor, nos presenta las primeras 18 fuentes y cada línea contiene la dirección del host que contiene la fuente, el nombre de la fuente y el precio de realizar una búsqueda. En la penúltima línea vemos la petición de las palabras clave que queremos buscar y finalmente, la última línea resume las órdenes básicas de *swais*.

#### ☞ EJECUCION DE *wais*

El plan general para usar *wais* es seleccionar una o más fuentes y entonces realizar una búsqueda, para lo cual hay que moverse hasta ella y pulsar la BARRA ESPACIADORA, *wais* indicará que la fuente ha sido seleccionada marcándola con un carácter, podemos seleccionar tantas fuentes como queramos antes de empezar a buscar, en cualquier momento, podemos seleccionar una nueva fuente moviéndonos a ella y pulsando la BARRA ESPACIADORA. Si decidimos dejar de usar una fuente que está seleccionada, nos movemos a ella y pulsamos de nuevo la BARRA ESPACIADORA, con lo que eliminamos la selección de la fuente. Para eliminar la selección de todas las fuentes y empezar de nuevo, hay que pulsar "=" (signo igual). Como recomendación antes de empezar una nueva búsqueda es bueno eliminar la selección de fuentes de la búsqueda anterior.

Una vez seleccionadas una o más fuentes, ya se puede realizar una búsqueda. Para empezar hay que pulsar RETURN. El cursor se moverá a la línea próxima al final de la pantalla en donde debemos escribir las palabras clave que queramos y finalmente hay que pulsar RETURN para comenzar la búsqueda. *wais* comprobará cada fuente especificada y mostrará una lista de posibilidades, *wais* organizará la lista de forma que los mejores resultados aparezcan primero, y aparece también una puntuación que nos da una idea de la forma en que encaja un elemento con las palabras clave, la puntuación más alta es de 1000. Ahora, visto lo que *wais* ha encontrado, hay que decidir si queremos profundizar en algo, para lo cual basta con seleccionarlo y pulsar





RETURN. *wais* se conectará con el servidor apropiado y pedirá una copia del archivo real. Cuando llegue, *wais* usará un programa de paginación para mostrar el archivo, pantalla a pantalla.

Mientras leemos el archivo, la pantalla es controlada por el programa de paginación. En este punto, hay dos órdenes básicas que se pueden usar. Para ver la siguiente pantalla de texto, hay que pulsar la BARRA ESPACIADORA. Para terminar de visualizar el artículo, hay que pulsar *q*. Mientras leemos el artículo, pulsando *q* salimos del programa de paginación, pero en cualquier otro elemento, pulsando *q* terminamos *wais*.

Si no estamos satisfechos con los resultados de la búsqueda, podemos hacer otro intento pulsando *w*. Esto nos lleva de nuevo a la línea de entrada de palabras clave, podemos cambiar las palabras clave y comenzar una nueva búsqueda. Cuando decidimos que hemos terminado de buscar en ese conjunto de fuentes, hay que pulsar *s* con lo que llegamos nuevamente a la pantalla de fuentes. Y hasta que se realicemos una nueva búsqueda, podemos visualizar los resultados de la búsqueda anterior pulsando *r*.

Con algunos clientes *wais*, se puede seleccionar todo o parte de un elemento como representativo del tipo de cosas que buscamos, se le puede decir a *wais* que guarde todas las palabras de un elemento como ayuda para búsquedas futuras, es lo que definimos como pertinencia realimentada. Pero cuando se hace *telnet* a un cliente *wais* de acceso público, la pertinencia realimentada no se tiene en consideración, sin embargo si usamos un cliente *wais* sofisticado en nuestra propia computadora, podemos usar la pertinencia realimentada para ayudarnos a crear búsquedas personalizadas de gran eficiencia.

A título de resumen mostramos los comandos básicos de *swais*.



## Comandos básicos de *swais*

Ordenes básicas	
h	mostrar un sumario de órdenes (ayuda)
?	igual que h
q	salir de <i>swais</i>

Mover la línea activa	
Flecha Abajo	mover una línea abajo
j	igual que Flecha Abajo
CTRL-N	igual que Flecha Abajo
Flecha Arriba	mover una línea arriba
k	igual que Flecha Arriba
CTRL-P	igual que Flecha Arriba
número	saltar al número de línea especificado
/patrón	saltar a la fuente que comience con el patrón especificado
J	mover a la siguiente pantalla
CTRL-D	igual que J
K	mover a la pantalla anterior
CTRL-U	igual que K

Seleccionar una fuente	
BARRA ESPAC.	seleccionar/no seleccionar una fuente
=	eliminar todas las selecciones de fuentes
RETURN	(después de seleccionar fuentes) preguntar nuevas palabras clave
r	volver a mostrar el resultado de la búsqueda anterior
v	mostrar información técnica sobre una fuente



Realizar una búsqueda	
RETURN	(después de introducir palabras clave) empezar a buscar
RETURN	(mientras se examinan los resultados de una búsqueda) mostrar elemento
w	preguntar nuevas palabras clave
s	volver a mostrar la pantalla de fuentes

Leer un artículo	
BARRA ESPAC.	mostrar la siguiente pantalla
q	terminar de leer el artículo

### ☞ TRUCOS PARA ELEGIR UNA FUENTE

Lo que determina la eficiencia de *wais* para encontrar recursos, es la selección de fuentes que nosotros hagamos, he aquí algunos consejos para seleccionar la fuente adecuada. El primer truco es hacer una búsqueda preliminar de la fuente *directory-of-servers* usando palabras clave que describan la fuente que pensamos que nos ayudará. Hay que escoger qué tipo de referencia es probable que contenga la información que deseamos. El segundo truco supone un ahorro real de tiempo. Podemos obtener un sumario de las fuentes de *wais* por *ftp anónimo*. Este sumario contiene pequeñas descripciones de todas las fuentes de *wais*, recopiladas por categorías. Una forma fácil y rápida de elegir una fuente es observar este archivo, encontrar la categoría que encaja con nuestras necesidades y usar cualquier fuente que nos parezca bien. Por otra parte, lo anterior nos da una buena visión global de todo lo que *wais* puede acceder. La única advertencia es que el sumario está con frecuencia anticuado; probablemente habrá nuevas fuentes que *wais* conoce, pero que no están todavía categorizadas.

Se puede cargar el sumario de fuentes de *wais* vía *ftp anónimo* de *kirk.Bond.edu.au*. Hay que buscar en el directorio llamado */pub/Bond\_Uni/doc/wais* un archivo llamado *src-list.txt*. Un host alternativo de *ftp anónimo* es *archive.orst.edu*, donde el mismo archivo está almacenado en el directorio llamado */pub/doc/wais*.



## ☰ LISTAS DE CORREO

### ☞ INTRODUCCION

Las listas de correo están basadas en una idea muy simple, cuando enviamos un mensaje de correo a otro usuario, especificamos una dirección. Si queremos enviar un mensaje a más de una persona, es posible definir un nombre especial. llamado *alias*, que representa a un grupo de personas.

Por ejemplo, supongamos que se define un alias llamado *ejecutivos* para representar la dirección de tres personas llamadas Pablo, Alfredo y Susan. Siempre que se envíe un mensaje de correo a *ejecutivos*, el programa de correo lo enviará automáticamente a cada uno de esos tres usuarios. Supongamos que Pablo tiene una idea que desea compartir con los otros, lo que tiene que hacer es enviar un mensaje a *ejecutivos* y todos recibirán una copia, ahora, si Alfredo quiere hacer algún comentario en relación al mensaje de Pablo envía un mensaje a *ejecutivos*, con lo que el mensaje se envía a todos los usuarios de la lista. Esto es un grupo de discusión de tres usuarios.

Ahora, pensemos en este mecanismo de discusión a gran escala, imaginemos un alias que contenga la dirección de correo de decenas o quizás centenares de usuarios, repartidos por todo **INTERNET**, cualquier mensaje que se envíe al alias se distribuirá automáticamente a todos los componentes del grupo. Los usuarios pueden hablar, argumentar, ayudar a otros, discutir problemas, compartir información, etc. Cualquier cosa que un miembro del grupo escriba será recibido por todos los componentes del grupo.

Esto es una *lista de correo*. Como usuarios de **INTERNET**, nosotros tenemos acceso a varios miles de estas listas, cada una dedicada a un tema específico. Y es diferente al concepto de *grupos de noticias* o *grupos de discusión* que tratamos anteriormente.

La primera gran diferencia es que para participar en *Usenet* son necesarios muchos más conocimientos, por ejemplo es necesario saber utilizar un programa de lectura de noticias, mientras que con una lista de correo, todo lo que necesitamos es saber cómo enviar y leer mensajes de correo, algo que se debe aprender de todas formas. Esto significa que aquellos usuarios que no tienen acceso a *Usenet* pueden aún participar en grupos de discusión.



Otra diferencia es que las discusiones en una lista de correo llegan en forma de mensajes al buzón de correo personal. Algunas listas sólo envían unos pocos mensajes al día, sin embargo, es muy común que una lista muy activa genere decenas de mensajes todos los días. Puesto que estos mensajes se depositan en el buzón de correo personal, es preciso hacer algo con ellos. Los artículos de *Usenet*, en cambio, se almacenan en una ubicación central en la red y son gestionados por el administrador del sistema. Cuando se participa en listas de correo, en ocasiones es normal encontrar cientos de mensajes en el buzón.

Una de las principales ventajas de las listas de correo es que se pueden seleccionar las que más nos interesan y recibir los mensajes automáticamente siendo sólo necesario leer el correo, con *Usenet*, hay que ejecutar un programa de lectura de noticias y comprobar los grupos de noticias favoritos cada vez que se quiere ver si ha llegado algo nuevo. Además, la mayoría de los administradores de sistemas eliminan automáticamente los artículos después de un determinado período de tiempo. Sin embargo, *Usenet* es más conveniente para otras cosas, ya que podemos participar en un grupo cuando queramos en el momento que lo deseemos, mientras que en una *lista de correo* es necesario enviar un mensaje especial de correo para ser incluido en una lista y otro mensaje para ser eliminado de la lista.

## TIPOS DE LISTAS DE CORREO

Cada lista de correo tiene un administrador, y la mayoría de los casos, se trata de una persona a la que se hace referencia como gestor de la lista, administrador o coordinador. La principal función del administrador es mantener la lista de direcciones actualizada.

Todas las listas de correo tienen una dirección oficial, en la mayoría de las listas, cada mensaje que se envía a esa dirección se pasa automáticamente a cada uno de los componentes de la lista, es decir que cualquiera puede participar con sólo enviar un mensaje a la dirección apropiada. Cuando se recibe un mensaje de la lista, éste se envía desde la dirección de la lista de correo, y si contestamos el mensaje usando la orden *reply*, la respuesta se enviará de nuevo a todos los componentes de la lista, por lo que antes de responder a un mensaje, hay que considerar si es mejor enviar una respuesta privada a la persona que envió el mensaje, ya que en algunas ocasiones es mejor enviar una respuesta privada que un mensaje que podrán leer todos los componentes de la lista.



Algunas listas de correo son *moderadas*, lo que significa que todos los mensajes se envían a una persona llamada *moderador* (que en algunos casos no es la misma persona que el administrador). El moderador decide qué artículos deben enviarse a todos los miembros de la lista y realizan algunas tareas básicas de edición y organización del material que se recibe.

Algunos moderadores organizan los mensajes dentro de colecciones denominadas *recopilaciones*, las cuales son parecidas a un número de una revista electrónica: una serie completa de mensajes y artículos organizados de una forma fácil de leer. Algunos moderadores incluirán una tabla de contenido junto con la recopilación para facilitar la lectura de los mensajes más interesantes. La principal ventaja de una lista de correo moderada es que sólo se reciben los mensajes más interesantes (en opinión de alguien), mientras que muchas listas *no moderadas* tienen una gran cantidad de mensajes aburridos y redundantes que habrá que vadear para encontrar las verdaderas joyas.

La principal desventaja de una lista moderada es que su mantenimiento genera una gran cantidad de trabajo y la única compensación que reciben los moderadores es el sentimiento de gratitud que proviene de aquellos que ayudan a sus compañeros. (Por esta razón, la mayoría de las listas de correo son no moderadas.)

La parte más importante de la administración de una lista de correo es mantener actualizada la lista de personas que la componen, cuando solicitamos ser incluidos en una lista de correo, se dice que nos estamos *subscribiendo* en esa lista, y cuando solicitamos ser eliminados de la lista, se dice que estamos *eliminando la suscripción*.

Hay dos formas básicas en las que se administran las listas de correo. Algunas listas son mantenidas por un programa, para subscribirse enviamos un mensaje a una dirección especial, todos los mensajes enviados a esta dirección se procesan automáticamente por el programa de administración de la lista y puesto que el mensaje será leído por un programa, hay que utilizar un formato particular. Otras listas en cambio, son mantenidas por una persona, enviamos un mensaje que es leído por el administrador, quien añade o elimina manualmente de la lista a los usuarios, aunque la mayoría de las listas son públicas, existen algunas privadas a las que no es posible subscribirse sin permiso del administrador.



## ☞ LISTA DE LISTAS

Para conocer qué listas de correo hay disponibles existen una serie de fuentes donde podemos obtener un sumario de listas de correo, éstos se conocen como *listas-de-listas*.

Una de las lista-de-listas más grandes es mantenida por SRI International en el Network Information System Center (NISC) de Menlo Park, California. Y el propósito de esta lista es proporcionar una descripción de cada lista de correo pública -Bitnet e **INTERNET**- junto con las instrucciones para realizar la suscripción.

Hay dos formas de obtener la lista-de-listas SRI, en primer lugar, puede cargarse (copiarla a nuestra computadora) vía *ftp anónimo* desde *ftp.nisc.sri.com*, hay que buscar en el directorio */netinfo* un archivo llamado *interest-groups.Z*, y descomprimirlo. La segunda forma de obtener esta lista, consiste en enviar un mensaje a la dirección *mail-server@nisc.sri.com*. En el cuerpo del mensaje, debe escribirse *send netinfo/interest-groups*, recibiremos la lista en forma de varios mensajes, por lo que deberemos de guardarlos por separado y unirlos posteriormente.

Otra gran lista-de-listas es mantenida por la Dartmouth University de New Hatnpshire. Al igual que la lista SRI, la lista Dartmouth también suministra información de cada lista existente en Bitnet e **INTERNET**. Los administradores de la lista de Dartmouth editan la lista para eliminar duplicaciones y aquellas listas sólo de interés local. Antes de copiar esta lista, debería leerse un archivo que contiene instrucciones, para lo que debe utilizarse *ftp anónimo* para conectarnos con el host *dartcmsl.dartmouth.edu*. En el directorio llamado *siglists* (hay que usar *cd siglists*) debe copiarse el archivo de instrucciones llamado *read.me* (hay que usar *get read.me*). En este archivo, se pueden leer las distintas formas de obtener la lista-de-listas. Si se quiere obtener utilizando *ftp*, hay que usar la orden *get listserv.lists*.

Cuando se está interesado en un tema en particular y se desea encontrar una lista de correo perteneciente a esa área de conocimiento, existe otro camino para obtener y examinar una larga serie de archivos. Una forma fácil es utilizar *wais*, lo que hay que hacer es buscar la fuente denominada *mailing-lists*, la cual contiene varias listas-de-listas de Bitnet e **INTERNET** así como una lista de los grupos de *Usenet*.



Existe una lista-de-listas que contiene información de todas las listas de correo de **INTERNET**. Esta lista se actualiza regularmente y se envía a los grupos de *Usenet news.lists*, *news.announce.newusers* y *news.answers*. También se puede obtener este archivo vía *ftp anónimo* en *rtfm.mit.edu*. Hay que buscar los archivos llamados *part1*, *part2*, *part3*, ...en el directorio */pub/usenet/news.answers/mail/mailling-lists*.

## ☞ SUBSCRIPCIÓN A UNA LISTA DE CORREO

Algunas listas de **INTERNET** están administradas por un software de administración y debemos consultar la documentación para subscribirnos, sin embargo la mayoría de las listas son administradas por una persona. En estos casos, es necesario enviar un mensaje a esta persona solicitando que nos añada (o elimine) en la lista. La principal información que debemos conocer es donde hay que enviar la petición. Cada lista de correo de **INTERNET** tiene una dirección con el formato estándar de **INTERNET**, por ejemplo existe una lista de correo de **INTERNET** llamada *musicals*, dedicada al teatro musical, y su dirección es *musicals@world.std.com*.

Por convenio, todas las peticiones administrativas para listas de **INTERNET** deben ser enviadas a una dirección similar a la de la lista, pero añadiendo la palabra *"-request"* al identificador de usuario. Por ejemplo, hay una lista llamada *disney-comics* dedicada a la discusión de los personajes y libros de Disney, la dirección de esa lista es *disney-comics@student.docs.uu.se*. Para subscribirse, hay que enviar un mensaje a:

*disney-comics-request@student.docs.uu.se*.

Veamos cómo hacer una subscripción a la lista de correo *musicals*:

<p><i>mail musicals-request@world.std.com</i> <b>Subject:</b> <i>subscription request</i> <i>Please subscribe me to the musicals mailing list.</i> <i>Thank you.</i> <i>Pedro Huerta</i>      <i>pedro.huerta@industry.net</i></p>
--

Nunca debemos enviar una petición de subscripción a la dirección normal de una lista de correo, ya que de hacerlo todos los miembros de la lista recibirán la petición, por lo que hay que enviar la petición a la dirección administrativa especial.





## ☞ INTRODUCCION

Como ya sabemos, no hay una autoridad central que organice a **INTERNET**, realmente hay tantas computadoras en **INTERNET** que nadie puede seguir el rastro de todas ellas. En consecuencia, no es sorprendente que no haya algo así como un directorio central de usuarios de **INTERNET**.

Como vimos en el capítulo 2, si conocemos el nombre, apellido y dirección de una persona podemos buscarla a través de *finger*, pero que pasa si la persona que andamos buscando está en una máquina no-Unix, o su servidor no tiene ese servicio. En esta sección mostraremos algunas estrategias que podemos seguir para localizar a alguien dentro de **INTERNET**.

Un *directorio de páginas blancas* es un servicio que permite buscar en una base de datos el nombre de alguien o su dirección de correo electrónico, los directorios de páginas blancas frecuentemente contienen también otra información, tal como la dirección postal, un nombre de departamento, un número de teléfono, etc. En la mayoría de los casos, usar un directorio de páginas blancas es fácil, ya que lo que tenemos que hacer es introducir un nombre y si está en el directorio, se nos mostrará toda la información conocida acerca de ese nombre.

El nombre "directorio de páginas blancas" fue creado para dar la imagen de una guía telefónica por computadora. Sin embargo, la analogía no es del todo cierta, ya que todas las guías telefónicas funcionan de la misma manera y contienen la misma información estándar, mientras que en **INTERNET**, hay una gran variedad de directorios de páginas blancas y no todos funcionan de la misma forma.

### ☞ *gopher*

Si conocemos a alguien de alguna organización, especialmente si es de una universidad, el mejor sitio para empezar es *gopher*, ya que muchas organizaciones tienen un servidor de nombres CSO o algún otro tipo de directorio de páginas blancas, por lo tanto, hay que empezar buscando un servidor *gopher* de esa organización.

Si se encuentra uno, hay que buscar un elemento de un menú que contenga algún tipo de directorio, tal como un servidor de nombres CSO o un elemento que se refiera a una lista telefónica. Otra alternativa es conectar con el



Si se encuentra uno, hay que buscar un elemento de un menú que contenga algún tipo de directorio, tal como un servidor de nombres CSO o un elemento que se refiera a una lista telefónica. Otra alternativa es conectar con el servidor *gopher* de la Universidad de Notre Dame, en el estado de Indiana (la dirección es *gopher.nd.edu*) en el menú principal, hay que seleccionar las fuentes de información de *Non-Notre Dame*, después la opción *Phone Books -- Other Institutions/* y veremos un menú que ofrece varias categorías de directorios de páginas blancas (que ellos llaman "servidores de directorio"). Desde aquí hay acceso a cientos de directorios de páginas blancas diferentes, no sólo de universidades, sino de todo tipo de organizaciones.

## ☞ SERVIDORES WHOIS

Un servidor *whois* es otro tipo de directorio de páginas blancas que permite consultar una base de datos de nombres y direcciones de correo electrónico, normalmente de una organización en particular. Hay un gran número de servidores *whois* en **INTERNET**. Se puede obtener una lista de servidores *whois* vía *ftp anónimo*. Para encontrar los servidores de *ftp anónimo* que tienen esta lista, hay que usar *archie* y realizar una búsqueda *sub* de *whois-servers*.

Hay dos formas principales de acceder a un servidor *whois*. Primera, nuestro sistema puede tener la orden llamada *whois*. De ser así, se escribe el nombre de la orden, seguido de *-h*, seguido del nombre de un servidor *whois* determinado, seguido del nombre de la persona que estamos buscando.

La segunda forma de usar un servidor *whois* es a través del acceso público de *telnet*, y podremos utilizarlo con el identificador de usuario *whois*. Y cuando se vea el indicativo *Whois*: hay que introducir el nombre que andamos buscando. El más conocido es *whois.internic.net* por lo hay que hacer *telnet* a él.

## ☞ KNOWBOT

Un *knowbot* es un programa automático que busca información inteligentemente, por ejemplo podríamos decirle que busque donde sea una descripción del Ferrari Devil y él buscaría en los lugares apropiados de **INTERNET**.



Una vez establecida la conexión, podemos solicitar ayuda con el símbolo `?`, la órden básica es *query* seguida del nombre que andamos buscando.

#### ☞ FRED Y LOS DIRECTORIOS X.500

X.500 es un directorio internacional de páginas blancas que fué desarrollado para proporcionar un sistema universalmente aceptado para guardar información sobre personas, en la práctica todavía no ha sido aceptado de forma universal, pero existe ya un gran número de organizaciones que han adoptado este esquema, y se puede buscar en sus directorios usando el programa *Fred*.

Para ello, hay que hacer *telnet* a *wp.psi.com* o bien a *wp2.psi.com* y conectarse con el identificador de usuario *fred*. Una vez establecida la conexión, podemos obtener ayuda con la orden *help* o *manual*.

#### ☞ EL SERVIDOR DE DIRECCIONES DE *Usenet*

La computadora *rtfm.mit.edu* es el almacén principal de los archivos de *Usenet*. Por ejemplo, contiene copias de todas las listas de preguntas más frecuentes (FAQ). Esta computadora es administrada por Jonathan Kamens, quien ha puesto en marcha un importante directorio de páginas blancas llamado *servidor de direcciones de Usenet*. Un programa en *rtfm.mit.edu* rutinariamente examina cada artículo que es enviado a *Usenet* y busca la línea de cabecera *From:*, extrae todas las líneas de todos lo artículos y guarda su contenido en una base de datos. Así, si la persona que buscarnos ha enviado un artículo de *Usenet* en el último año, hay bastantes posibilidades de encontrarlo.

Para usar este servicio, hay que enviar un mensaje a:

*mail-server@rtfm.mit.edu*

Hay que dejar la línea *Subject:* en blanco, en el cuerpo de la carta hay que escribir una línea con el siguiente formato:

*send usenet-addresses/nombre*



Donde *nombre* describe a la persona que buscamos. El programa buscará en la base de datos de *Usenet* y devolverá todas las líneas que contengan el patrón especificado. En consecuencia, se puede usar un nombre, un apellido o un identificador de usuario. Además, no es necesario usar letras mayúsculas ya que el programa no distingue entre mayúsculas y minúsculas.

Otra forma de usar el servidor de direcciones de *Usenet* es vía *wais*, con lo que permite buscar directamente en la base de datos. Todo lo que se necesita es ejecutar *wais* y seleccionar la base de datos llamada *usenet-addresses*. Lo bueno de usar *wais* es que la búsqueda puede ser más compleja, por ejemplo, se puede especificar más de una palabra en la búsqueda y *wais* clasificará las respuestas en orden de pertinencia.

## NETFIND

*Netfind* es un programa que explora activamente a **INTERNET**, buscando una computadora que conozca a la persona que estamos intentando encontrar. O bien, buscando la ubicación de una computadora de la cual conocemos su domicilio. Para usar *Netfind*, necesitamos tener una idea de donde está la persona, y la ventaja de *Netfind* es que no sólo busca un nombre y una dirección, sino que intenta encontrar la información de *finger* sobre esa persona. Si la persona parece tener más de una dirección de correo, *Netfind* hará una recomendación de cual es la mejor a utilizar.

Hay dos formas de usar *Netfind*. Primero podemos tener el programa *Netfind* en nuestra computadora. O segundo, podemos hacer *telnet* a un host público con *Netfind* y utilizar desde ahí el programa.

Si tenemos un programa *Netfind* en nuestra computadora, podemos ver la documentación del manual interactivo de Unix -*Netfind* sólo funciona bajo sistemas Unix- usando la orden *man*. Típicamente comenzaremos con introducir la instrucción *Netfind* seguida del nombre de la persona que buscamos, seguido de alguna localidad (universidad, institución o empresa). A lo que el programa nos responderá el resultado de su búsqueda.

Cuando *Netfind* empieza una búsqueda, verifica en la parte de **INTERNET** que le hemos especificado y comprueba cuantas computadoras hay, si hay más de unas pocas, *Netfind* nos dirá que hay demasiados "dominios", entonces listará las computadoras más prometedoras y nos pedirá que elijamos en cuales queremos buscar.



Cuando *Netfind* empieza una búsqueda, verifica en la parte de **INTERNET** que le hemos especificado y comprueba cuántas computadoras hay, si hay más de unas pocas, *Netfind* nos dirá que hay demasiados "dominios", entonces listará las computadoras más prometedoras y nos pedirá que elijamos en cuales queremos buscar.

Cuando encuentra a una persona, pide la información de *finger* y nos la muestra, y de ser posible *Netfind* mirará las fechas en que esa persona se ha conectado en cada computadora y deducirá cual de las varias direcciones de correo (en los casos en que tengamos varias direcciones) sería mejor utilizar.

Si no tenemos un programa *Netfind*, podemos hacer *telnet* a uno de los servidores públicos de *Netfind* que se enlistan a continuación

<i>Localización</i>	<i>Dirección INTERNET</i>	<i>Dirección IP</i>
<i>Australia</i>	<i>archie.au</i>	<i>139.130.4.6</i>
<i>Canadá</i>	<i>macs.ce.mcgill.ca</i>	<i>132.206.61.15</i>
<i>Chile</i>	<i>malloco.ing.puc.cl</i>	<i>146.155.1.43</i>
<i>República Checa</i>	<i>Netfind.vslib.cz</i>	<i>147.230.16.1</i>
<i>Inglaterra</i>	<i>monolith.cc.ic.ac.uk</i>	<i>155.198.5.3</i>
<i>Singapur</i>	<i>lincoln.technet.sg</i>	<i>192.169.33.6</i>
<i>Eslovaquia</i>	<i>nic.uakom.sk</i>	<i>192.108.131.12</i>
<i>Corea del Sur</i>	<i>nic.nm.kr</i>	<i>143.248.1.100</i>
<i>USA: Alabama</i>	<i>redmont.cis.uab.edu</i>	<i>138.26.64.4</i>
<i>USA: Colorado</i>	<i>bruno.cs.colorado.edu</i>	
<i>USA: Minnesota</i>	<i>mudhoney.niicro.umn.edu</i>	<i>134.84.132.7</i>
<i>USA: Texas</i>	<i>Netfind.oc.com</i>	<i>192.82.215.92</i>
<i>USA: Virginia</i>	<i>ds.internic.net</i>	<i>198.49.45.10</i>
<i>Venezuela</i>	<i>dino.conicit.ve</i>	<i>150.188.1.10</i>

*conectarse con el identificador de usuario netfind.*



# INTERNET



## Tipos de Archivos

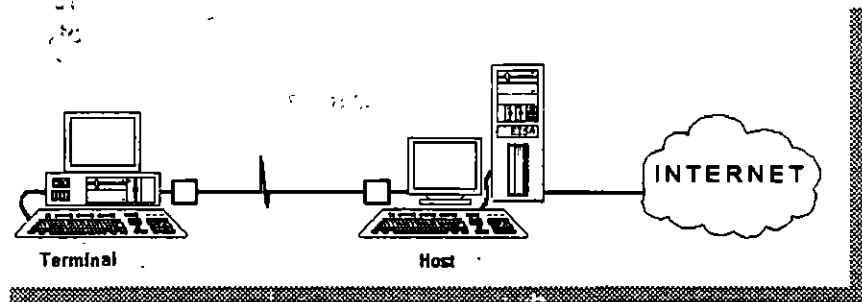
- ☐ **Compactados.**
  - ↳ ( Unix: compress, uncompress, \*.z )
  - ↳ ( Dos: pkzip, pkunzip, \*.zip )
  
- ☐ **Recopilados.**
  - ↳ ( Unix: tar, \*.tar, \*.tar.z )
  - ↳ ( Dos: , \*.taz )
  
- ☐ **Texto o Binarios.**
  - ↳ ( binario a texto: uuencode )
  - ↳ ( texto a binario: uudecode, \*.uue, \*.uue.z )
  
- ☐ **Tipo shar.**
  - Ejecutable que se desempaqueta.
  - ↳ ( \*.shar, \*.shar.z )
  - ↳ ( sh, unshar )

Notas:

# INTERNET



## Transferencia de Archivos



Notas:

# INTERNET



## Transferencia de Archivos

☐ Comandos Unix, emulando terminal

- ↳ sz.- Carga un archivo desde un Host remoto
- ↳ rz.- Descarga un archivo a un Host remoto

☐ Utilizan el protocolo *zmodem*

Ejemplos:

```
sz -a document
sz -b picture.gif
rz -a document
```

- a → tipo texto.
- b → tipo binario.

Notas:



# INTERNET



---

## Lista de Correos

- ↵ Se agrupan personas en una lista de un tema de interés común, para el manejo de mensajes de correo electrónico.
- ↵ Se define un Alias, para un grupo de personas.

Notas:

# INTERNET



## Servicio *wais*

### Wide Area Information Service

- ↳ Servicio de información de área extensa.
- ↳ Hace búsquedas en base a palabras.
- ↳ Arquitectura cliente/servidor.

Notas: