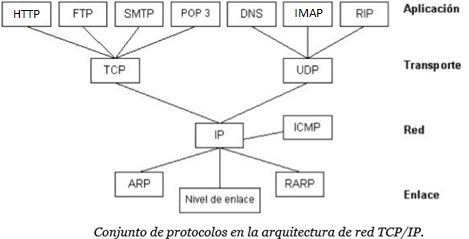
***Unidad 13. Protocolos de la Capa de Aplicaciones.***

Tema 1: Protocolo DNS. Protocolo de Servicio de Nombres de Dominio.

Tema 2: Protocolo HTTP. Protocolo de Transferencia de Hiper Texto.

Tema 3: Protocolo FTP. Protocolo de Transferencia de Archivos.

Tema 4: Protocolos del Correo de Internet. Email.



***Tema 1: Protocolo DNS. Protocolo de Sistema de Nombres de Dominio.***

**Qué significa DNS**

Cada equipo conectado directamente a Internet tiene al menos una [dirección IP](https://es.ccm.net/contents/267-direccion-ip) específica. Sin embargo, los usuarios no desean trabajar con direcciones numéricas, como por ejemplo **194.153.205.26**, sino con un nombre de dominio, por ejemplo: **es.ccm.net.**  
  
 Es posible asociar nombres en lenguaje normal con direcciones numéricas gracias a un sistema llamado **DNS** (*Domain Name System*, sistema de nombres de dominio). Ésta correlación entre las direcciones IP y el nombre de dominio asociado se llama **resolución de nombres de dominio**.

**Nombre de 'host'**

En los inicios de [TCP/IP](https://es.ccm.net/contents/282-tcp-ip), como las redes no eran muy extensas, o en otras palabras el número de equipos conectados a la misma red era bajo, los administradores de red crearon archivos llamados **tablas de conversión manual**. Estas tablas de conversión eran archivos secuenciales, por lo general llamados **hosts** o **hosts.txt**, y asociaban en cada línea la dirección IP del equipo con el nombre literal relacionado, denomi-nado **nombre de host**, con la ubicación:

**C:\Windows\System32\Drivers\etc\hosts**

**Introducción al sistema de nombres de dominio**

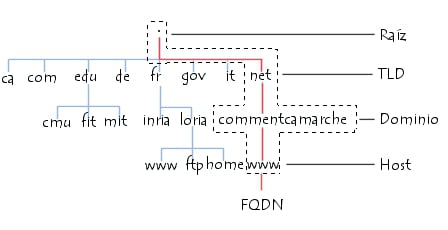
El anterior sistema de tablas de conversión exigía una actualización manual de las tablas para la totalidad de los equipos en caso de incluir o modificar el nombre de una máquina. Con el aumento en tamaño de las redes y sus interconexiones, fue necesario implementar un sistema de gestión para los nombres que fuese jerárquico y fácil de administrar. El sistema llamado **sistema de nombres de dominio** (**DNS**) fue desarrollado en noviembre de 1983 por Paul Mockapetris.

Este sistema ofrece un **espacio de nombre** jerárquico que permite garantizar:

1. la singularidad de un nombre en una estructura arbórea, es decir el nombre es único y expresado en forma arbórea, [www.google.com.ar](http://www.google.com.ar).
2. Un sistema de **servidores de distribución** que permite que el **espacio de nombre** esté disponible.
3. Un sistema de **cliente** que permite "resolver" nombres de dominio, es decir, interrogar a los servidores para encontrar la dirección IP que corresponde a un nombre.

**Espacio de nombre**

La estructura del sistema DNS se basa en una estructura arbórea en donde se definen los dominios de nivel superior (llamados TLD, **dominios de nivel superior**); esta estructura está conectada a un nodo raíz representado por un punto.



FQDN,  (nombre de dominio totalmente calificado).

Cada nodo del árbol se llama **nombre de dominio** y tiene una etiqueta con una longitud máxima de 63 caracteres.

Por lo tanto, todos los nombres de dominio conforman una estructura arbórea inversa en donde cada nodo está separado del siguiente nodo por un punto (“.”).

**www.commentcamarche.net.**

El extremo de la bifurcación se denomina *host*, y corresponde a un equipo o entidad en la red. El nombre del ordenador que se provee debe ser único en el dominio respectivo, o de ser necesario, en el subdominio. Por ejemplo, el dominio del servidor web por lo general lleva el nombre **www**.

La palabra **dominio** corresponde formalmente al sufijo de un nombre de dominio, es decir, la recopilación de las etiquetas de nodo de la estructura arbórea, con excepción del ordenador.

**commentcamarche.net.**

El nombre absoluto está relacionado con todas las etiquetas de nodo de una estructura arbórea, separadas por puntos y que termina con un punto final y se denomina **dirección FQDN,** (nombre de dominio totalmente calificado). La profundidad máxima de una estruc-tura arbórea es 127 niveles y la longitud máxima para un nombre FQDN es 255 caracteres. La dirección FQDN permite ubicar de manera única un equipo en la red de Internet.

**Servidores de nombres de dominio**

Los equipos llamados **servidores de nombres de dominio** permiten establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red. Cada dominio cuenta con un servidor de nombre de dominio, llamado **servidor de nombre de dominio principal**, así como también un **servidor de nombre de dominio secundario**, que puede encargarse del servidor de nombre de dominio principal en caso de falta de disponibilidad.

Cada servidor de nombre de dominio está especificado en el servidor de nombre de dominio en el nivel superior inmediato, lo que significa que la autoridad sobre los dominios puede delegarse implícitamente. El sistema de nombre es una arquitectura distribuida, en donde cada entidad es responsable de la administración de su nombre de dominio.

Los servidores relacionados con los dominios de nivel superior, (TLD), se lla-man **servidores de dominio de nivel superior**. Son 13, están distribuidos por todo el mundo y sus nombres van desde ***a.root-servers.net***hasta ***m.root-servers.net*.**

El servidor de nombre de dominio define una zona, es decir, una recopilación de dominios sobre la cual tiene autoridad. Si bien el sistema de nombres de dominio es transparente para el usuario, se deben tener en cuenta algunos aspectos. Cada equipo debe configurarse con la dirección de un equipo que sea capaz de transformar cualquier nombre en una dirección IP. Este equipo se llama servidor de nombres de dominio. Cuando te conectas a Internet, el proveedor de servicios automáticamente modificará los parámetros de tu red para hacer que estos servidores de nombres de dominio estén disponibles. La dirección IP de un segundo **servidor de nombres de dominio,** (servidor de nombres de dominio secundario), debe también definirse: el servidor de nombres de dominio secundario puede encargarse del servidor de nombres de dominio principal en caso de fallas en el sistema.

**Resolución de nombres de dominio**

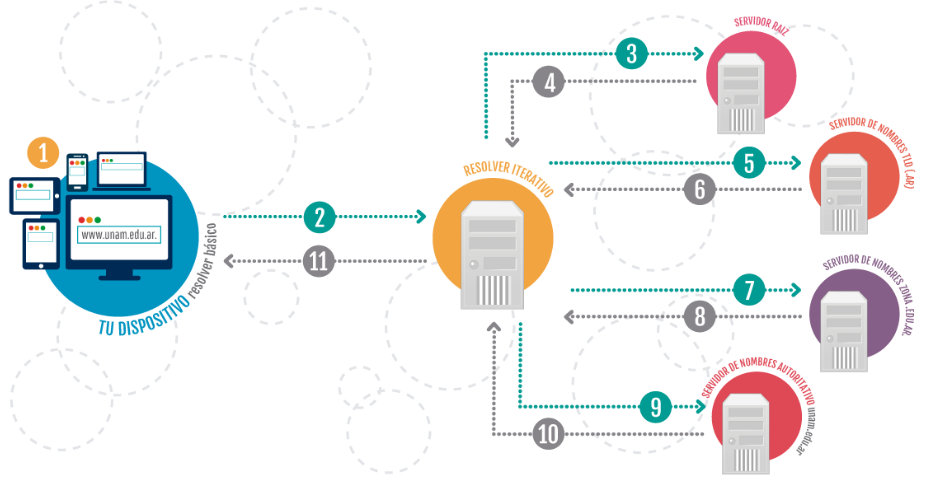
La aplicación cliente DNS que consiste en encontrar la dirección IP relacionada al nombre de un ordenador se conoce como **resolución del nombre de dominio**.

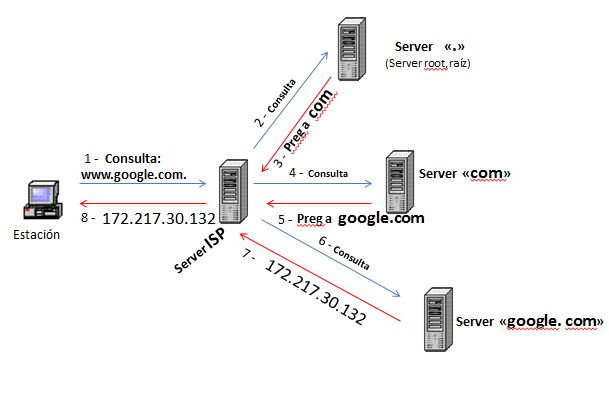
Cuando una aplicación desea conectarse con un *host* conocido a través de su nombre de dominio, (por ejemplo, **www.google.com.**), ésta interroga al servidor de nombre del proveedor del servicio de internet, ISP. Uno principal y el otro secundario, siempre hay redundancia por posibilidades de fallos.

Entonces se envía una solicitud al primer servidor de nombre de dominio, (llamado el servidor de nombre de dominio principal). Si este servidor de nombre de dominio tiene el registro en su caché, lo envía a la aplicación; de lo contrario, interroga a un servidor de nivel superior, (en nuestro caso **“.”**). El servidor de nombre de nivel superior envía una lista con las direcciones IP de los servidores de nombres de dominio con autoridad sobre el dominio consultado **google.com.**, (principal y secundario).

Entonces el servidor de nombres de dominio principal será interrogado y devolverá el registro correspondiente al siguiente dominio del servidor (en nuestro caso **.com**).

Y así se continúa iterativamente consultando los Servidores de dominio, descendiendo por el sistema de nombres arbolado.

****



**Tipos de registros**

Por lo general, un registro de DNS contiene la siguiente información:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de dominio (FQDN)** | **TTL** | **Tipo** | **Clase** | **RData** |
| es.ccm.net. | 3600 | A | IN | 163.5.255.85 |

**Nombre de dominio**: el nombre de dominio debe ser un nombre FQDN,  (nombre de do-minio totalmente calificado), es decir completo, debe terminar con un punto.

**TTL:** es el tiempo de vida de esta información, contabilizada en segundos.

**Tipo**: El tipo de recurso puede ser uno de los siguientes:

* **A**: éste es el tipo que hace coincidir el nombre de dominio con la dirección IPv4.
* **CNAME**: permite definir un alias para el nombre de dominio. Es útil para sumi-nistrar nombres alternativos relacionados con diferentes servicios en el mismo equipo. Por ejemplo un server de Web y un server FTP, en el mismo equipo físico.
* **MX** (*Mail eXchange*): es el que suministra el servidor de correo electrónico.

**Clase**: la clase es **IN** (relacionada a protocolos de Internet).  
  
**RDATA**: Aquí se encuentra la información esperada:

* Para A: la dirección IP de 32 bits.
* Para CNAME: el nombre alternativo de dominio.
* Para MX: nombre del servidor de correo electrónico.



Ubicación Geográfica de los Servidores DNS

***Tema 2: Protocolo HTTP. Protocolo de Transferencia de Hiper Texto.***

**HTTP** (Hypertext Transfer Protocol) es el protocolo que gobierna todas las conexiones entre un servidor y un navegador.

El Protocolo de Transferencia de HiperTexto, (Hypertext Transfer Protocol), es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web, (Navegadores de Internet: Internet Explorer, Google Chrome, Firefox, etc.), y los servidores WEB. Ambos se comunican por medio del protocolo HTTP. Fue propuesto por Tim Berners-Lee.



Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la siguiente manera:

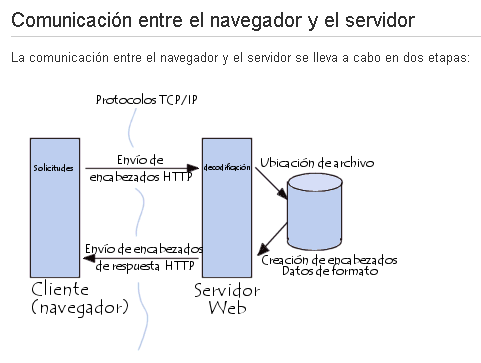
Un proceso servidor escucha en un puerto de comunicaciones TCP, (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

El protocol HTTP se basa en sencillas operaciones de solicitud/respuesta:

Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su resultado. Todas las operaciones adjuntan un recurso.

Cada recurso Web, (documento HTML, archivo multimedia o aplicación CGI), es conocido por su URL. (Uniform Resource Locator,   
Localizador Uniforme de Recursos).

Las aplicaciones CGI, **(Common Gateway Interface),** fueron una de las primeras prácticas de crear [contenido dinámico](file:///F:\wiki\Contenido_din%25C3%25A1mico) para las [páginas web](file:///F:\wiki\P%25C3%25A1gina_web).



[***http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html***](http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html)

**Etapas de una transacción HTTP.**

Para profundizar más en el funcionamiento del protocolo HTTP, veremos primero un caso particular de una transacción HTTP.

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

* Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
* El cliente Web analiza la URL, separando sus diferentes partes:

1. El protocolo de acceso, (HTTP o HTTPS).
2. La dirección IP del servidor WEB.
3. El PuertoTCP, (HTTP: Puerto 80 o HTTPS: Puerto 443).
4. El recurso solicitado al servidor, (documento HTML, archivo multimedia o aplicación CGI).

* A solicitud del cliente se establece una conexión TCP entre el cliente y el servidor en el Puerto 80 para HTTP o 443 para HTTPS.
* El cliente realiza la petición enviando:

1. El comando necesario, (GET, POST, HEAD, etc.).
2. La dirección del recurso requerido, (el contenido de la URL que sigue a la dirección del servidor).
3. La versión del protocolo HTTP empleada, (por ejemplo: HTTP/1.0).
4. Otras datos de información, que incluyen información sobre las capaci-dades del cliente navegador, y algunos datos opcionales para el servidor.

* El servidor responde al cliente:

1. Con un código de estado, indicando comunicación correcta, incorrecta, etc.
2. El tipo de dato **MIME** que envía, (texto, audio, video, etc.).
3. El contenido de la información solicitada. El archivo solicitado.

**\* MIME:** ***Multipurpose Internet Mail Extensions***, (extensiones multipropósito de correo de internet): conjunto de especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.).

* Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

El protocolo estándar HTTP/1.0, define tres comandos a utilizar por el cliente navegador:

1. Comando GET. Es utilizado por el Cliente Navegador para solicitar cualquier tipo de información almacenada en el Servidor WEB.

Se utiliza siempre que se pulsa sobre un enlace, o se teclea una URL.

Como resultado, el servidor WEB envía al Cliente Navegador el documento correspondiente a la URL solicitada.

1. Comando HEAD. Es utilizado por el Cliente Navegador para solicitar alguna información sobre un recurso almacenado en el Servidor WEB, por ejemplo tamaño, tipo, fecha de modificación del recurso.

Es utilizado por los gestores de cachés de páginas del Cliente Navegador o por los Servidores Proxy, para conocer cuándo es necesario actualizar la copia que se mantiene de un recurso.

1. Comando POST. Es utilizado por el Cliente Navegador para enviar información al Servidor WEB, por ejemplo los datos contenidos en un formulario. El Servidor WEB pasará esta información a un proceso interno encargado de su tratamiento.

**Códigos de respuesta del Servidor WEB.**

El código de respuesta es un número que indica que ha pasado con la petición:

* Códigos con formato 1xx: Respuestas informativas. Indica que la petición ha sido recibida y se está procesando.
* Códigos con formato 2xx: Respuestas correctas. Indica que la petición ha sido procesada correctamente.
* Códigos con formato 3xx: Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
* Códigos con formato 4xx: Errores causados por el cliente. Indica que ha habido un error en el procesado de la petición a causa de que el cliente ha hecho algo mal.
* Códigos con formato 5xx: Errores causados por el servidor. Indica que ha habido un error en el procesado de la petición a causa de un fallo en el servidor.

**Cabeceras HTTP**

Son los [datos](https://es.wikipedia.org/wiki/Metadato) que se envían en las peticiones o respuestas HTTP, para proporcionar información sobre la transacción en curso.

Cada cabecera es especificada por un nombre de cabecera seguido por dos puntos, un espacio en blanco y el valor de dicha cabecera seguida por un retorno de carro seguido por un salto de línea. Se usa una línea en blanco para indicar el final de las cabeceras. Si no hay cabeceras la línea en blanco debe permanecer.

Ejemplo:

GET /index.html HTTP/1.1

Host: www.example.com

Referer: www.google.com

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Firefox/45.0

Connection: keep-alive

[Línea en blanco]

Las cabeceras pueden tener metadatos que tienen que ser procesados:

1. Por el cliente (ej. se puede indicar el tipo del contenido del recurso que se envía, archivo, multimedio o Gif).
2. Por el servidor (ej. tipos de representaciones aceptables por el cliente del contenido que pide).
3. Por los intermediarios (ej. como gestionar la actualización del caché de memoria por parte de los [proxys](https://es.wikipedia.org/wiki/Servidor_proxy)).

Dependiendo del tipo de mensaje en el que puede ir una cabecera las podemos clasificar en cabeceras de petición, cabeceras de respuesta y cabeceras que pueden ir tanto en una petición como en una respuesta.

Podemos clasificar las cabeceras según su función. Por ejemplo:

* Las que indican las capacidades aceptadas por el Cliente Navegador que envía el mensaje:

**Accept** indica el [MIME](https://es.wikipedia.org/wiki/MIME) aceptado. **Accept-Charset** indica el código de caracteres aceptado. **Accept-Encoding** indica método de compresión aceptado.

**Accept-Language** indica el idioma aceptado.

**User-Agent**  para describir al cliente.

* Las que describen el contenido:

**Content-Type**  indica el [MIME](https://es.wikipedia.org/wiki/MIME) del contenido.

**Content-Length**  longitud del mensaje.

* Las que sirven para autentificación:

**WW-Authenticate** solicita autenticación.

* Las que refieren a la URLs:

**Referer**  Indica el origen de la petición.

* Las que ahorran transmisiones**:**

**Date**  fecha de creación.

**Ejemplo de diálogo HTTP**,

Para obtener un recurso con el [URL](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto#URL)  <http://www.example.com/index.html>

1. Se abre una conexión en el puerto 80 del host www.example.com. El puerto 80 es el puerto predefinido para HTTP.
2. Se envía un mensaje en el estilo siguiente:

GET /index.html HTTP/1.1

Host: www.example.com

Referer: www.google.com

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Firefox/45.0

Connection: keep-alive

[Línea en blanco]

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web.

***Respuesta del servidor:***

Envía el Servidor Web Muestra el Cliente Navegador



**HTTP/2.**

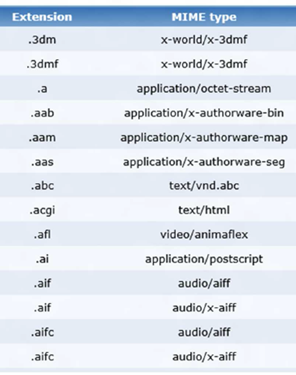
En HTTP/2, se incorporó un mecanismo para realizar la transferencia de «streams» múltiples sobre una misma conexión TCP. Así, cada recurso puede ser solicitado y entregado sobre un stream, y los recursos que requieren más tiempo para ser entregados no interfiere con la entrega de otros recursos más ligeros.

Para poder realizar la carga de todos los elementos de una página sobre una única conexión, la especificación HTTP/2 recomienda que las implementaciones realizadas en navegadores y servidores web permitan al menos 100 streams por conexión.



\*MIME. ***Multipurpose Internet Mail Extensions***, (extensiones multipropósito de correo de internet): conjunto de especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.).

La siguiente imagen muestra los distintos tipos MIME con sus correspondientes extensiones de archivos:



***Tema 3: Protocolo FTP. Protocolo de Transferencia de Archivos.***

La transferencia de archivos entre Cliente FTP y servidor FTP, puede producirse en cualquier dirección. El Cliente puede pedir o enviar un fichero al Servidor.

El usuario Cliente FTP debe identificarse ante el Servidor FTP, para acceder a los archivos que contiene el Servidor FTP.

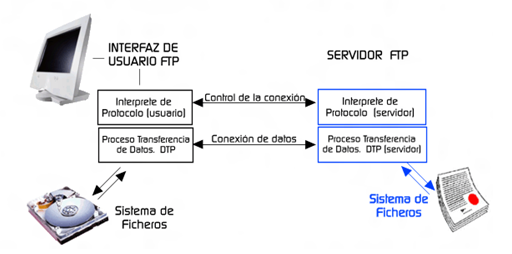
El Servidor FTP es responsable de autenticar al Cliente FPT antes de permitir la transferencia de archivos.

Para establecer una sesión FTP, se debe establecer una conexión TCP, o sea orientada a conexión y completamente segura. Se emplean dos conexiones: la primera es para el login, (autenticación), y la segunda es para gestionar la transferencia de datos.

Como es necesario hacer un login en el Server FTP remoto, el usuario debe tener un nombre de usuario y una contraseña para acceder a los archivos y directorios. El usuario que inicia la conexión asume la función de Cliente FTP, mientras que la estación remota adopta la función de Servidor FTP.

En ambos extremos de la sesión FTP se construye lo siguiente:

1. Una interfaz de usuario.
2. Un Intérprete de Protocolo, (PI).
3. Un proceso de transferencia de datos.



La interfaz de usuario se comunica con el Intérprete de Protocolo, (PI), que está a cargo del control de la conexión. Éste Intérprete de Protocolo comunica la información necesaria a su propio sistema de archivos.

En el otro extremo de la conexión, el Intérprete de Protocolo, (PI), además de su función de responder, inicia la conexión de datos.

Durante la transferencia de archivos, los Procesos de Transferencia de Datos, (DTPs), se ocupan de gestionar la transferencia de datos.

Una vez que la operación se ha completado, el Intérprete de Protocolo, (PI), cierra la conexión de control.

Para ejecutar una transferencia de archivos, el usuario en el Cliente FTP, comienza haciendo un login en el Server FTP. El usuario debe ingresar un Identificador y una Contraseña para ingresar al Server FTP, a menos que el Server FTP permita el ingreso con un login anónimo.

En este último caso el usuario sólo tiene que usar el Identificador “anonymous” y la contraseña que se le indique durante el proceso de login. Las contraseñas pueden ser: 1) ninguna, 2) guest, 3) E-mail del usuario.

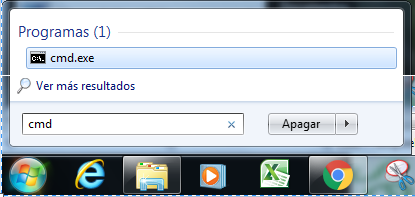
Los comandos necesarios para la transferencia de archivos son los siguientes:

* **Open:** Selecciona el Server FTP remoto, e inicia la sesión con el login.
* **User:** Identificación del usuario Cliente FTP.
* **Pass:** Contraseña del usuario Cliente FTP.
* **Get:** Selecciona el archivo a ser transferido del Server al Cliente FTP.
* **Put:** Selecciona el archivo a ser transferido del Cliente al Server FTP.

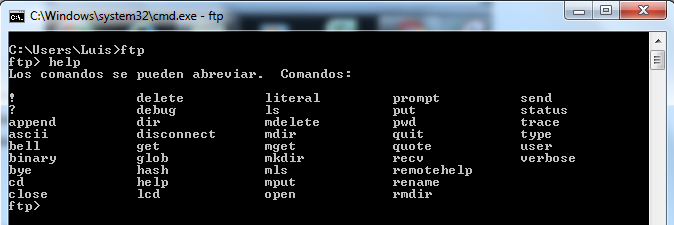
Existen muchos comandos que pueden utilizarse en una sesión FTP.

**Ejemplo de FTP en una sesión DOS.**

Ingresamos a una sesión DOS, ejecutando el comando **“cmd”**, en la ventana de ejecución de Windows:



Dentro de la sesión DOS, ingresamos al ambiente FTP, con el comando FTP, y dentro del ambiente FTP ejecutamos el comando HELP, y nos muestra todos los comandos del ambiente FTP.



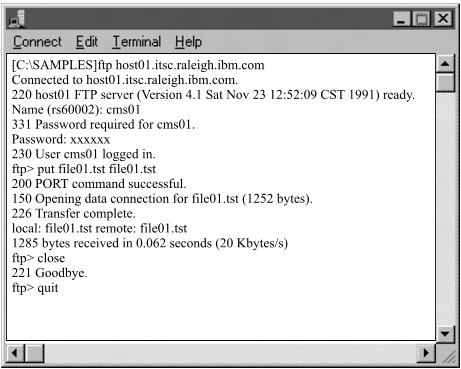
Cuando ingresamos al ambiente FTP, el prompt que es **“C:\>”** se transforma en **“ftp>”** para indicarnos el cambio de ambiente.

Para salir del ambiente FTP se ejecuta el comando “**quit”** que nos devuelve la sesión DOS.

Ejemplo de la transferencia del archivo “**welcome.msg”** solicitada por el Cliente FTP al Server FTP, en la página siguiente:



En el siguiente ejemplo el cliente FTP transfiere un archivo con el comando **“put”** al Server FTP.



Existen aplicaciones que usando los comandos anteriormente vistos permiten la transferencia de archivos, desde un entorno gráfico, por ejemplo FILEZILLE.



***Tema 4: Protocolos del Correo de Internet. Email.***

***Funcionamiento del Correo Electrónico y funciones que cumplen los siguientes protocolos vinculados al mismo.***

**Protocolo SMTP**: Protocolo Simple de Transferencias de Correos.

(Simple Mail Transfer Protocol).

**Protocolo POP3**: Protocolo de Oficina de Correos, versión 3.

(Post Office Protocol-3)

**Protocolo IMAP**: Protocolo de Acceso a Correo de Internet.

(Internet Mail Access Protocol)

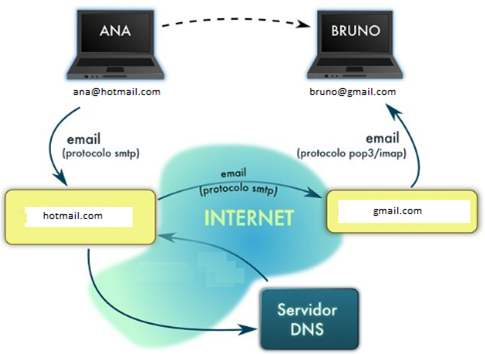
**Descripción del funcionamiento del Correo Electrónico.**

El correo electrónico es uno de los principales usos que le damos a Internet, ya que es un medio de comunicación gratuito, rápido y confiable. Con unos cuantos clicks por ejemplo, podemos enviar el mismo mensaje a toda nuestra oficina, incluyendo imágenes y otros archivos, de manera sencilla y prácticamente instantánea.

Así mediante Clientes Web, es decir navegadores como por ejemplo Google Chrome, Internet Explorer, etc., sólo tenemos que entrar a la página de Internet que co-rresponda a nuestro Cliente de Correo, Hotmail.com, Gmail.com, etc., y listo.

En la infraestructura de Correo Electrónico en Internet, se requiere de una serie de procesos que garantizan la llegada correcta de nuestro correo electrónico a otras personas.

La siguiente figura ilustra el esquema general del funcionamiento del Correo Electrónico explicado mediante un ejemplo.



En primera instancia tenemos a dos usuarios, Ana y Bruno. Pensemos que Ana, (cuya dirección de correo electrónico es ana@hotmail.com), quiere enviar un correo electrónico a Bruno, (cuya dirección es [bruno@gmail.com](mailto:bruno@gmail.com)).

La segunda parte de la dirección de correo, (después de @), corresponde al proveedor de correo, Hotmail.com, Gmail.com, etc.

La primera parte, (antes de @), corresponde al nombre de la persona dentro del proveedor de correo, Hotmail.com, Gmail.com, etc.

Ana abre entonces su programa de correo electrónico, (Hotmail.com), y una vez autenticada, es decir, habiendo introducido su nombre de usuario y contraseña, envía su correo.

El destino es el Servidor de Correo Electrónico “Gmail.com”, entonces el Servidor de Correo Electrónico “Hotmail.com”, envía una consulta a un servidor DNS, el cual confirma la dirección IP del Correo electrónico “Gmail.com”.

Esta información se conoce como el registro MX del servidor de correo de “Gmail.com”. Usualmente, se tiene más de un registro MX, de forma que si algún Servidor de Correo deja de funcionar, pueda seguir el proceso con algún otro.

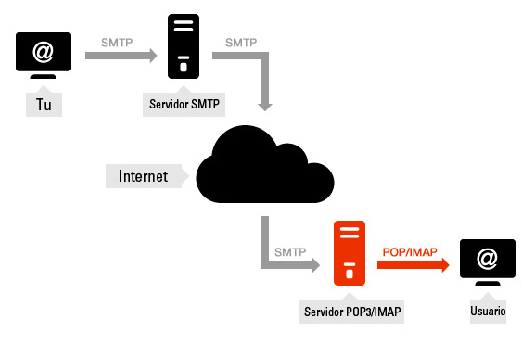
Una vez conocido este dato, el Servidor de Correo “Hotmail.com” contacta al Servidor de Correo “Gmail.com” y de estar disponible, le enviará nuestro correo electrónico, en donde residirá hasta que Bruno abra su cliente y lea su correo.

Este procedimiento se realiza mediante el protocolo SMTP, cuyas siglas en inglés significan Simple Mail Transfer Protocol, o Protocolo Sencillo de Envío de Correo.

El SMTP soporta una gran cantidad de opciones; un Servidor de Correo puede devolver mensajes de error, por ejemplo si no tiene registrado al usuario que buscamos, o si Ana se equivocara al escribir la dirección. En ese caso Ana recibiría un correo electrónico indicándole el problema.

Por otra parte, también es común que un servidor de correo tenga instalado un antivirus y un filtro para distinguir al spam, o correo no deseado.

**Protocolos SMTP, POP3 e IMAP**



***SMTP (Simple Mail Transfer Protocol)***

El protocolo SMTP posee limitaciones en cuanto a la recepción de mensajes del Servidor de Correo. Por esta razón su función es únicamente enviar el correo desde el Cliente SMTP Emisor al Servidor SMTP Receptor, para que posteriormente haga su trabajo el POP3 e IMAP.

|  |
| --- |
| Durante el Flujo de transacción normal de correo SMTP, todos los comandos, réplicas y datos intercambiados son líneas de texto de código ASCII de siete bits, delimitadas por un <CRLF>.  Todas las réplicas tienen un código numérico al comienzo de la línea.   * El Cliente SMTP Emisor establece una conexión TCP en el puerto 25 con el Servidor SMTP Receptor, que se encuentra escuchando en el puerto 25. El Servidor SMTP Receptor responde con el mensaje "220 Service ready", o "421 Service not available", cuando el destinatario es temporalmente incapaz de responder. * El Cliente SMTP Emisor envía un HELO, (abreviatura de "hello"), y el Servidor SMTP Receptor responde con su nombre de dominio. * El Cliente SMTP Emisor inicia ahora una transacción enviando el comando MAIL al Servidor SMTP Receptor. Si se acepta, el receptor responde con un "250 OK". * El Cliente SMTP Emisor mediante el comando RCPT TO, envía el destino del mensaje, si hay más de un destino deben agregarse comandos RCPT TO. El Servidor SMTP Receptor responderá con el mensaje "250 OK", si el servidor conoce el destino, o un "550 No such user here", si no es así. * El Cliente SMTP Emisor envía un comando DATA para notificar al Servidor SMTP Receptor que a continuación se envían los contenidos del mensaje. El Servidor SMTP Receptor responde con "354 Start mail input, end with <CRLF>.<CRLF>". Se trata de la secuencia de terminación que el Cliente SMTP Emisor debe usar para terminar los datos del mensaje. * El Cliente SMTP Emisor envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF>, y El Servidor SMTP Receptor responde con "250 OK" o el mensaje de error apropiado si cualquier cosa fue mal.   Ahora hay varias acciones posibles:   * El Cliente SMTP Emisor no tiene más mensajes que enviar, entonces cerrará la conexión enviando un comando QUIT, y el Servidor SMTP Receptor responde con "221 Service closing transmission channel". * El Cliente SMTP Emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 2 para enviar un nuevo MAIL.   Ejemplo:    ***Ejemplo de diálogo entre Cliente y Servidor SMTP.***  Cliente SMTP Emisor Servidor SMTP Receptor |
| escuchando en puerto 25  Establece una conexión TCP en puerto 25    "220 Service ready"  HELO  smtp.gmail.com  MAIL  "250 OK"    RCPT TO: user1@gmail.com  "250 OK"    DATA  "354 Start mail input, end with <CRLF>.<CRLF>"  Hola  Nos encontramos 16:00 hs.  .  "250 OK"  QUIT  "221 Service closing transmission channel" |

***POP3 (Post Office Protocol)***

Es uno de los protocolos utilizados por clientes de email (Hotmail, Gmail, etc.), para recoger mensajes del servidor de email. Los mensajes son transferidos desde el servidor hacia lacomputadora local cuando el usuario se conecta al servidor.

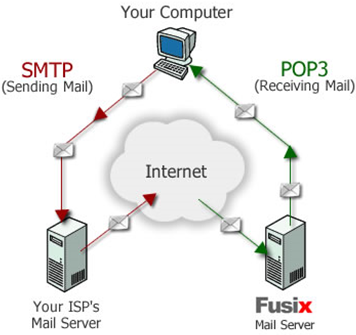
Después de recibir los mensajes, la conexión puede interrumpirse, procediéndose a la lectura de los mensajes sin necesidad de continuar conectado al servidor. Esto quiere decir que un computador después de haber recibido el mensaje correspondiente puede seguir con la lectura del mismo sin importar si sigue en existencia una conexión a la red.

Para establecer una conexión a un servidor POP, el cliente de correo abre una conexión TCP en el [puerto](https://www.ecured.cu/Puerto) 110 del servidor. Cuando la conexión se ha establecido, el servidor POP envía al cliente POP una invitación y después se envían entre sí otras órdenes y respuestas que se especifican en el protocolo. Como parte de esta comunicación, al cliente POP se le pide que se autentifique, (Estado de autenticación), donde el nombre de usuario y la contraseña del usuario se envían al servidor POP.

Si la autenticación es correcta, el cliente POP pasa al Estado de transacción, en este estado se puede utilizar [órdenes](https://www.ecured.cu/index.php?title=%C3%93rdenes&action=edit&redlink=1) LIST, RETR y DELE para mostrar, descargar y eliminar mensajes del servidor, respectivamente. Los mensajes definidos para su eliminación no se quitan realmente del servidor hasta que el cliente POP envía la orden QUIT para terminar la sesión. En ese momento, el servidor POP pasa al Estado de actualización, y se eliminan los mensajes marcados.

Algunas órdenes

* User (nombre): Identificación de usuario.
* Pass (password): Envía la [clave](https://www.ecured.cu/Clave) al servidor.
* Stat: Número de mensajes no borrados en el buzón y su longitud total.
* List: Muestra todo los mensajes no borrados con su longitud.
* Retr (número): Solicita el envío del mensaje especificando el número (no se borra del buzón).
* Top (número) (líneas): Muestra la cabecera y el número de líneas del mensaje.
* Dele (número): Borra el mensaje especificando el número.
* Rset: Recupera los mensajes borrados (en la conexión actual).
* Quit: Salir.



***IMAP (Internet Message Access Protocol).***

A diferencia del POP3, utilizando IMAP la conexión entre la computadora local y el servidor de email debe estar siempre activa pues hay una constante interacción entre ambos.

Los mensajes se mantienen en el servidor de email, aunque el usuario accede como si estuvieran localmente. Esta opción es útil para las personas que leen sus e-mails en diferentes computadoras.

El Cliente IMAP establece una conexión TCP en el puerto 143 con el Server IMAP, se autentica, y luego continúa con la sesión establecida, enviando órdenes y recibiendo respuestas, en un diálogo semejante al visto en el protocolo POP3, aunque contiene muchos más comandos que él.

Algunos comandos:

*SEARCH*, *FETCH*, *STORE, APPEND* para trabajar sobre los mensajes.

*LIST,* *SELECT,* *CREATE,* *RENAME,* *DELETE* para trabajar sobre las carpetas.

***Listado de todos los puertos que pueden utilizar los protocolos que intervienen en el correo electrónico, e-mail.***

Puertos de Servidores de Entrada:

* IMAP | Port TCP 993 (Transporte Seguro - función SSL habilitada)
* IMAP | Port TCP 143 (Transporte inseguro - Función SSL no habilitada)
* POP3 | Port TCP/UDP 995 (Transporte Seguro - función SSL habilitada)
* POP3 | Port TCP 110 (Transporte inseguro - Función SSL no habilitada)

Puertos de Servidores de Salida:

* SMTP | Port TCP 465 (Transporte Seguro - función SSL habilitada)
* SMTP | Port TCP 587 (Transporte inseguro. Conexión segura usando [STARTTLS](https://help.dreamhost.com/hc/en-us/articles/215612887#STARTTLS))
* SMTP | Port TCP 25 (Desactualizado y no recomendado).

Nota:

SSL significa Secure Sockets Layer, Capa de Conexiones Seguras, e indica que las transmisiones de datos son encriptados antes de ser enviadas, lo que aumenta la seguridad e inviolabilidad de la información

Fin de la Unidad 12.