

	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

## INTRODUCCIÓN A HTTP

**World Wide Web** (o la "**Web**") es un sistema de documentos de *hipertexto* enlazados y accesibles a través de *Internet*. Debe entenderse por *hipertexto*, la tecnología que permite organizar una base de información en bloques distintos de contenidos, conectados a través de una serie de hipervínculos o enlaces cuya activación o selección provoca la recuperación de información.

Con un *navegador Web*, un usuario visualiza *páginas Web* que pueden contener texto, imágenes u otros contenidos multimedia, y navega a través de ellas usando hipervínculos.

La visualización de una *página Web*, u otro recurso comienza normalmente tecleando la *URL* (Localizador Uniforme de Recursos o **Uniform Resource Locator**) de la página en el navegador Web, o siguiendo un enlace de hipertexto a esa página o recurso. Al aceptar, el primer paso en ejecutarse es la traducción de la parte del nombre del servidor de la URL en una dirección IP usando la base de datos distribuida de Internet conocida como DNS (Sistema de Nombres de Dominio o **Domain Name System**). Entonces el navegador establece una conexión TCP (Protocolo de Control de Transmisión o **Transmission Control Protocol**) con el servidor en esa dirección IP.

El siguiente paso es enviar una petición *HTTP* (Protocolo de Transferencia de Hipertextos o **HiperText Transfer Protocol**) al servidor Web solicitando el recurso. En el caso de una página Web típica, primero se solicita el texto *HTML* (Lenguaje de Marcas de Hipertextos o **HiperText Markup Language**) y luego es analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros archivos que formen parte de la página, en una rápida sucesión.

Entonces el navegador Web carga la página tal y como se describe en el código HTML, el CSS (Hojas de Estilo en Cascada o **Cascading Style Sheets**) y otros archivos recibidos, incorporando las imágenes y otros recursos si es necesario. Esto produce la página que ve el usuario en su pantalla.

### Cliente Web

Un *cliente Web* es un *navegador Web* o aplicación que permite al usuario recuperar y visualizar documentos de hipertexto a través de la comunicación con un servidor Web.

Entre los navegadores más conocidos encontramos:

- (1) Chrome
- (2) Internet Explorer
- (3) Mozilla Firefox
- (4) Opera
- (5) Safari

### Servidor Web

Un *servidor Web* es el programa que implementa el protocolo HTTP. Cabe destacar que por *servidor* se entiende tanto el programa como el equipo en que se ejecuta dicho programa.

Un servidor Web se encarga de mantenerse a la espera de *peticiones HTTP* llevadas a cabo por un cliente HTTP o navegador; este realiza una petición al servidor y este le responde con el contenido que el cliente solicita.

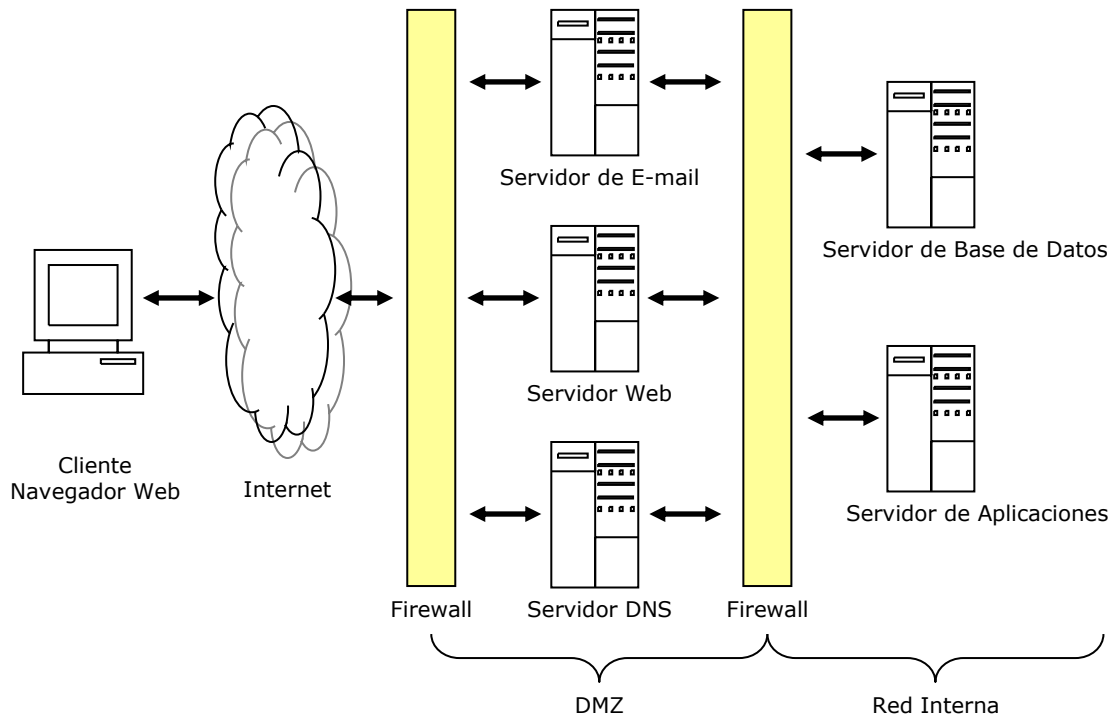
A través del uso de Virtual Host se permite que varias direcciones Web compartan el mismo servidor Web.

Entre los servidores Web más conocidos encontramos:

- (1) Apache: Programa *Open Source* que corre bajo sistema operativo UNIX, Macintosh o Windows.
- (2) IIS (**I**nternet **I**nformation **S**ervice): Esta directamente relacionado con equipos que funcionan con el sistema operativo Windows.

Un servidor Web se ubica dentro de la DMZ (Zona Desmilitarizada o **DeMilitarized Zone**) ya que necesitamos que tenga acceso desde afuera. La DMZ es una red local que se ubica entre la red interna de la organización y una red externa generalmente Internet.

Una posible estructura de red sería como la que se presenta a continuación, considerar que está armada desde el punto de vista lógico, sin entrar a los detalles físicos de conexión.



### Aplicaciones Web

Las *aplicaciones Web* son los fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:


- (1) **Aplicaciones en el lado del cliente:** El cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones Java (Applet) o Javascript: el servidor proporciona el código de las aplicaciones al cliente y este, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en el lenguaje Java y Javascript, aunque pueden añadirse más lenguajes mediante el uso de *plugins*.
- (2) **Aplicaciones en el lado del servidor:** el servidor Web ejecuta la aplicación; esta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.  
Entre los lenguajes o herramientas con los que se puede desarrollar una *aplicación Web* encontramos PHP, ASP, Perl, CGI, .NET y JSP.

### TCP/IP

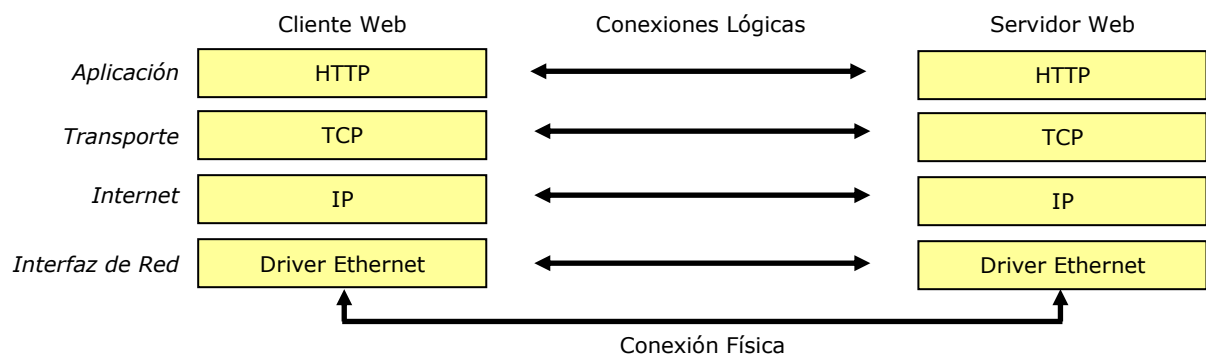
Un protocolo es un conjunto formal de reglas que deben ser seguidas para comunicarse. Un protocolo de bajo nivel define los detalles, tales como la velocidad de transmisión o los niveles de voltaje requerido para interpretar una señal como uno o cero. Por su parte, un protocolo de alto nivel define el formato de los datos como la secuencia de los mensajes a enviar.

TCP/IP es una colección de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Protocolo de Control de Transmisión o **T**ransmission **C**ontrol **P**rotocol) y el IP (Protocolo de Internet o **I**nternet **P**rotocol), que son los que dan nombre al conjunto. La arquitectura del TCP/IP consta de cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- (1) **Aplicación:** Se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de archivos (FTP), conexión remota (TELNET) y el protocolo HTTP (Hypertext Transfer Protocol).
- (2) **Transporte:** Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos. Donde:

	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

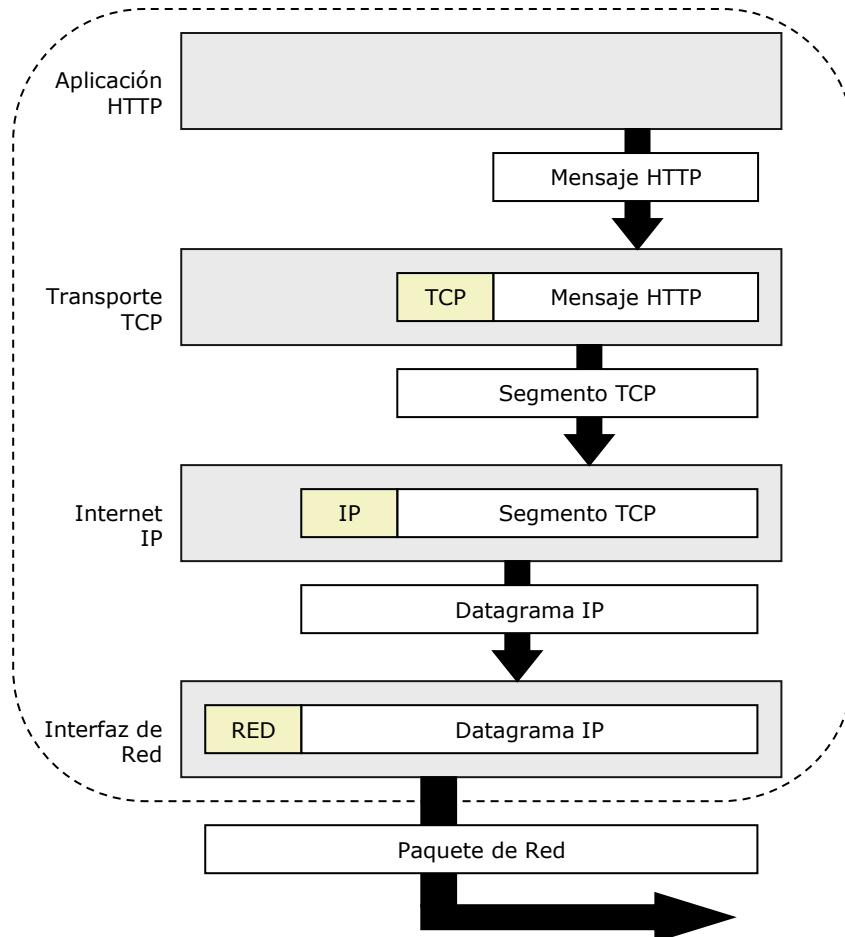
- a. **TCP** (*Protocolo de Control de Transmisión o **T**ransmission **C**ontrol **P**rotocol*): Utiliza el tipo de comunicación *connection-oriented*, es decir, que el protocolo se asegura que el dato transmitido es recibido en el orden correcto. Además, intercambia información de handshake (control) antes de iniciar la comunicación y recibe confirmaciones (acknowledgment) de la comunicación realizada; esto asegura el arribo de paquetes de datos.
  - b. **UDP** (*Protocolo de Datagrama de Usuario o **U**ser **D**atagram **P**rotocol*): Utiliza el tipo de comunicación *connection less*, es decir, no requiere handshake o acknowledgment por lo cual no asegura automáticamente la entrega de un paquete válido.
- (3) **Internet**: Es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.
- (4) **Interfaz de red**: Es la interfaz de la red real. TCP/IP no especifica ningún protocolo concreto, así que corre por las interfaces conocidas, como por ejemplo: Ethernet, IEEE 802.2, CSMA/CD, X.25, etc.



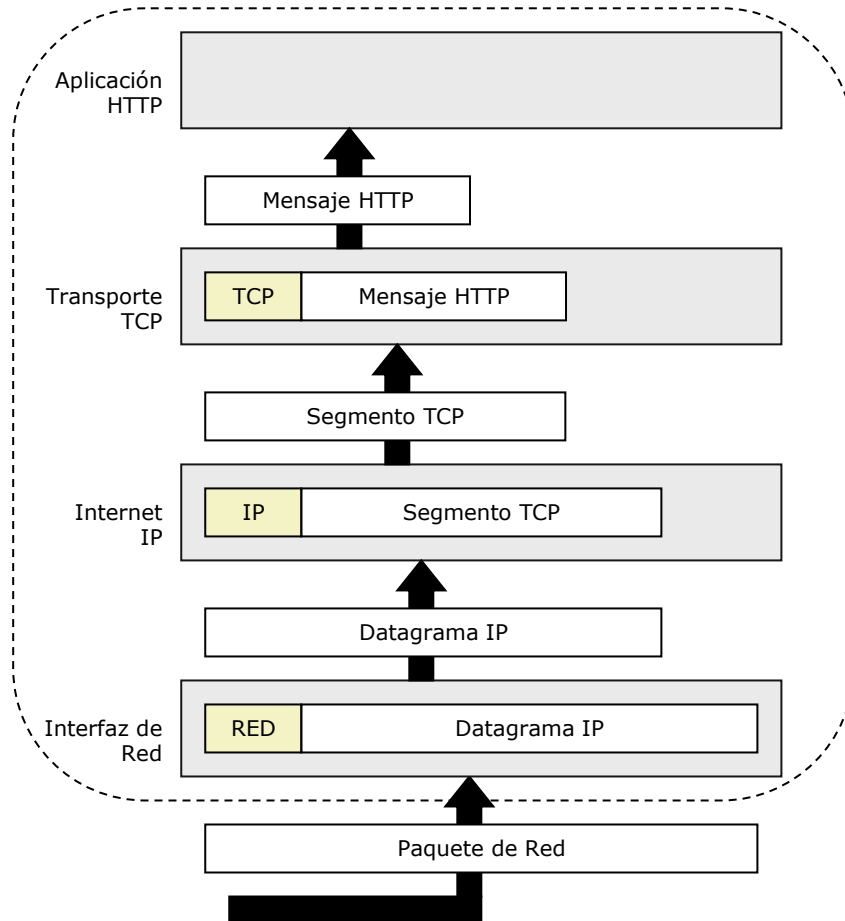
### Mensajes entre cliente y servidor Web


A continuación se ilustrará cómo viaja un mensaje a través de la red para el caso particular de una petición HTTP de un navegador Web al servidor Web correspondiente.

*Envío del Mensaje (Navegador)*



*Recepción del Mensaje (Servidor Web)*



	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

## URI vs. URL

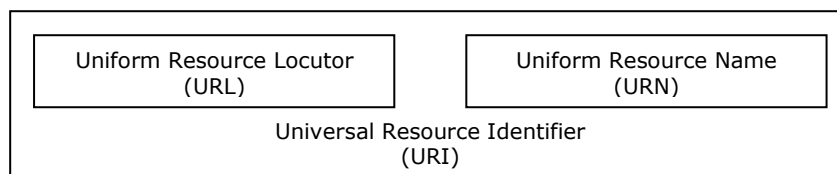
El Identificador Uniforme de Recurso o **Uniform Resource Identifier (URI)** es una secuencia compacta de caracteres que provee una forma simple y extensible para identificar un recurso de Internet, por recurso se entiende un servicio, página, documento, dirección de correo electrónico, etc. Dicho recurso es totalmente independiente del nombre con el que es conocido. Las partes de un URI son:

protocolo://username:password@host[:port]/path/file[?query] [#fragment]

En general es más familiar el Localizador Uniforme de Recurso o **Uniform Resource Locator (URL)** que es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar un recurso en la Web; técnicamente, un URL es un tipo particular de URI.

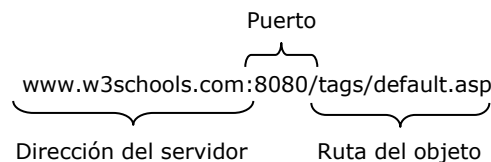
Otro tipo de URI, es el Nombre Universal de Recurso o **Uniform Resource Name (URN)** es un método de referenciar un objeto sin especificar su path completo al mismo. Globalmente, es similar al ISBN de un libro.

El mismo surgió en vistas de optimizar el acceso a ciertas páginas que son accedidas de manera constante y ante la necesidad de generar copias para reducir el tráfico de red. El problema es que los URLs no proporcionan ninguna manera de referirse a una página sin decir de manera simultánea dónde está. No hay forma de especificar "Necesito la página X y no me importa de dónde la obtengas". Por ello, se creó el URN para posibilitar la duplicación de páginas.



## Componentes del URL

- (1) Dirección del servidor
- (2) Número de puerto (opcional)
- (3) Ruta (path) del objeto



## Sintaxis del URL

protocolo://host[:port]/path [#seccion] [?query]

protocolo://username:password@host[:port]/path [#seccion] [?query]

query es un conjunto de parámetros separados por & (Ampersand)

## Codificación del URL

La codificación del URL comprime una cadena de caracteres imprimibles en un conjunto de caracteres ASCII. A su vez, reemplaza todos los caracteres no imprimibles y no seguros con signo % seguido de los dos dígitos hexadecimales del carácter ASCII correspondiente.

Nota: Un espacio puede ser representado por %20 o + (más)

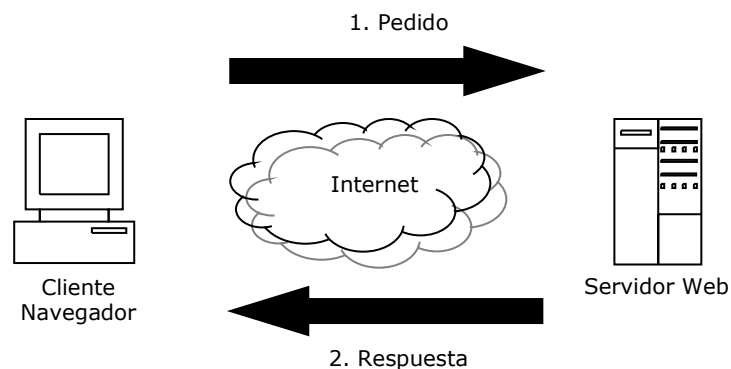
	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

## HTTP

El protocolo HTTP está basado en el modelo cliente-servidor. Un cliente HTTP abre una conexión y envía su solicitud al servidor, el cual responderá con el recurso solicitado (si está disponible y su acceso es permitido) y la conexión se cierra. Por lo cual opera bajo el paradigma solicitud/respuesta.

Utiliza el protocolo TCP/IP para establecer conexiones con el sistema remoto. Es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños archivos guardados en la propia máquina del cliente que puede leer un sitio Web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente.

Solo el cliente tiene la responsabilidad de iniciar la comunicación y es el único que puede hacerlo. Por lo cual, el cliente HTTP actúa y el servidor HTTP reacciona.



HTTP es el medio de transporte universal de Internet a través de firewalls. La versión actual de HTTP es la 1.1, y su especificación está en el documento RFC 2616. A su vez, dispone de una variante cifrada mediante SSL (Capa de Sockets Seguros o **Secure Sockets Layer**) llamada HTTPS, que es la versión segura del protocolo HTTP.

SSL construye una conexión segura entre los dos sockets (cliente – servidor), incluyendo:

- (1) Negociación de parámetros entre el cliente y el servidor.
- (2) Autenticación tanto del cliente como del servidor.
- (3) Comunicación secreta.
- (4) Protección de la integridad de los datos.

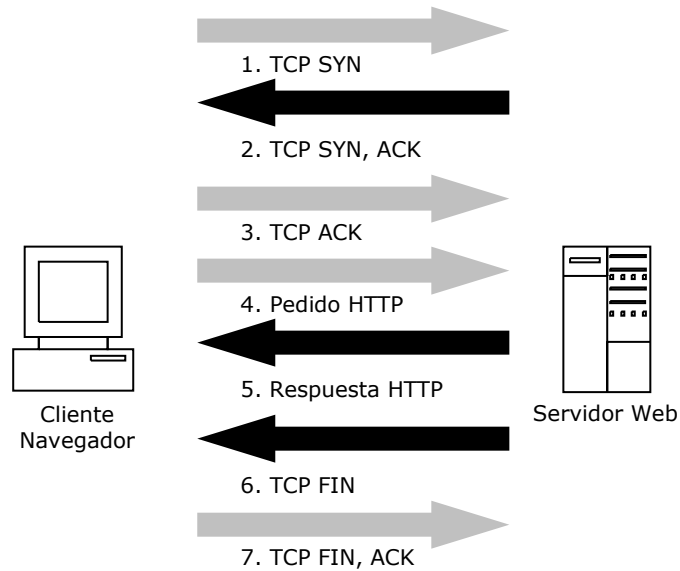
La posición de SSL en la pila de protocolos usuales del TCP/IP, es entre la capa de aplicación y la de transporte, el cual acepta solicitudes del navegador y envía al TCP para transmitir al servidor. Una vez que se ha establecido la conexión segura, el trabajo principal de SSL es manejar la compresión y encriptación. El HTTPS es el HTTP que se utiliza encima de SSL pero algunas veces está disponible en un nuevo puerto (433) en lugar del estándar (80). Además, el SSL no está restringido a utilizarse solo en navegadores Web pero es su aplicación más común.

### **Conexión entre el navegador y el servidor Web**

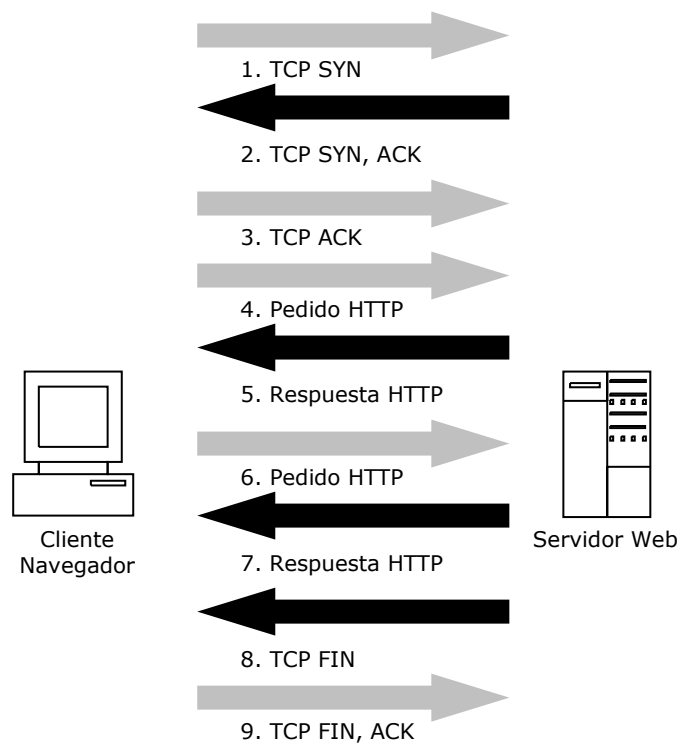
Una vez que el cliente Web conoce la dirección IP en la cual debe establecer su comunicación con el servidor Web, los pasos que siguen son:

- (1) El cliente abre una conexión al servidor empleando TCP, y por defecto al puerto 80.
- (2) El cliente envía un pedido al servidor mediante un comando HTTP.
- (3) El servidor responde el pedido, con un código de estado, varios campos y si es posible el contenido del recurso solicitado.
- (4) La comunicación es cerrada.
- (5) Se cierra la conexión TCP.

### Conexión No Persistente



### Conexión Persistente




### Pipeline

El cliente es quien tiene la decisión de emplear una conexión persistente. Cuando este establece una conexión persistente, no tiene que esperar la respuesta a un pedido para enviar otro pedido.

Solo es válido para los métodos GET y HEAD, no así para POST y PUT. Es importante tener en cuenta que el servidor debe soportar el protocolo HTTP 1.1 y estar configurado con una conexión keep-alive (conexión persistente).



	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

## Estructura del HTTP

El formato tanto del mensaje como de la respuesta es el siguiente:

```
<Línea inicial>
Encabezado-1: valor-1
...
Encabezado-n: valor-n
<Cuerpo del mensaje (Opcional)>
```

La línea inicial es diferente en las solicitudes y en las respuestas. En las solicitudes está formada por tres campos que se separan con un espacio en blanco: "Método Recurso Versión-del-Protocolo". Por ejemplo,

"GET / HTTP/1.1"

La línea inicial de una respuesta tiene tres campos separados por un espacio: "Versión-del-Protocolo Código-de-Respuesta Mensaje". Por ejemplo,

"HTTP/1.1 200 OK".


A la línea inicial pueden seguirle líneas adicionales que contienen más información. Estas son llamadas encabezados. Los encabezados están normados en el protocolo, e incluyen, en el caso de una solicitud, información del navegador y eventualmente del usuario cliente. En el caso de una respuesta, información sobre el servidor y sobre el recurso. Veamos el gráfico a continuación,

Nombre de cabecera pedida	Valor de cabecera pedida	Nombre de cabecera recibida	Valor de cabecera recibida
Host	localhost	Status	OK - 200
User-Agent	Mozilla/5.0 (Windows NT 6.0; rv:13.0) Gecko/20100101 Firefox/13.0.1	Date	Mon, 02 Jul 2012 02:51:28 GMT
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	Server	Apache
Accept-Language	es-ar;es;q=0.8,en-us;q=0.5,en;q=0.3	X-Powered-By	PHP/5.3.14
Accept-Encoding	gzip, deflate	Content-Length	2
Connection	keep-alive	Keep-Alive	timeout=5, max=100
		Connection	Keep-Alive
		Content-Type	text/html; charset=ISO-8859-15

Finalmente, el cuerpo del mensaje contiene el recurso a transferir o el texto de un error en el caso de una respuesta. En el caso de una solicitud, puede contener parámetros de la llamada archivos enviados al servidor.

HTTP define cuatro métodos u operaciones básicas y otras adicionales:

- (1) **GET**: Utilizado para solicitar un documento o recurso específico. Es la operación por defecto al navegar por la Web. Debe tenerse en cuenta que cuando se utiliza el método con propósitos de transferir datos, su uso es poco conveniente ya que los URLs tienen una limitación de 8.192 caracteres y muchas veces podría truncarse los datos que se envían al servidor.
- (2) **POST**: Empleado para transferir datos desde el cliente al servidor, los cuales serán procesados. El ejemplo clásico es el envío de datos de un formulario.
- (3) **PUT**: Utilizado para almacenar recursos en el servidor, por ejemplo, archivos.
- (4) **DELETE**: Empleado para borrar recursos del servidor.
- (5) **Operaciones Adicionales**:
  - a. **HEAD**: Es una operación especial que tan sólo nos recupera información del recurso, como el tamaño, la fecha de modificación, tipo, etc. Lo suelen utilizar los navegadores o servidores Proxy para comprobar el estado de su caché u otras operaciones. Es similar a la operación GET pero sin el cuerpo de la respuesta.
  - b. **TRACE**: Solicita que el cuerpo del mensaje sea retornado como se lo envió, es empleado principalmente para depuración o debugging.
  - c. **OPTIONS**: Proporciona una forma para que el cliente consulte al servidor sobre sus propiedades o las de un archivo específico.
  - d. **CONNECT**: Se utiliza para saber si se tiene acceso a un servidor, no necesariamente la petición llega al servidor. Este método se utiliza principalmente para saber si un proxy nos da acceso a un host bajo condiciones especiales.

	INGENIERÍA EN INFORMÁTICA – PLAN 2003 PROGRAMACIÓN DISTRIBUIDA Y COMPONENTES – 9º CUATRIMESTRE	
	APUNTE DE HTTP	VERSIÓN: 1.2 VIGENCIA: 10-03-2012

Los códigos de estado en la respuesta son:

- (1) **100 – 199**: Indican un mensaje informativo.
  - 111**: Conexión rechazada.
- (2) **200 – 299**: Indican éxito de alguna clase.
  - 200**: OK.
  - 201 – 203**: Información no oficial.
  - 204**: Sin contenido.
  - 205**: Contenido por recargar.
  - 206**: Contenido parcial.
- (3) **300 – 399**: Redirecciona el cliente a otra URL.
  - 301**: Mudado permanente.
  - 302**: Encontrado.
  - 303**: Vea otros.
  - 304**: No modificado.
  - 305**: Utilice un proxy.
  - 307**: Redirección temporal.
- (4) **400 – 499**: Indica un error del lado del cliente.
  - 400**: Solicitud incorrecta.
  - 401**: No autorizado.
  - 402**: Pago requerido.
  - 403**: Prohibido.
  - 404**: No encontrado.
  - 409**: Conflicto.
  - 410**: Ya no disponible.
  - 412**: Falló precondition.
- (5) **500 – 599**: Indica un error del lado del servidor.
  - 500**: Error interno.
  - 501**: No implementado.
  - 502**: Pasarela incorrecta.
  - 503**: Servicio no disponible.
  - 504**: Tiempo de espera de la pasarela agotado.
  - 505**: Versión de HTTP no soportada.

## MIME

Los tipos MIME (Extensiones Multipropósito de Correo Internet o **M**ultipurpose **I**nternet **M**ail **E**xtensions) son un estándar para el envío de información binaria a través de caracteres alfanuméricos, es decir, permite agregar una estructura al cuerpo del mensaje definiendo reglas de codificación para los mensajes no ASCII. Este estándar permite que, a través del protocolo HTTP (que maneja información en modo texto), podamos transferir archivos no textuales, como pueden ser imágenes, audio, vídeo, programas ejecutables etc.

Los tipos MIME definen grupos (antes del carácter "/") y tipos (después del carácter "/"). Así el tipo MIME "text/html" define a todos los archivos de texto que contienen código HTML, el tipo "video/mpeg" define a todos los archivos de vídeo almacenados en formato mpeg, etc. Para indicar cualquier tipo se puede utilizar el carácter "\*", tanto en el tipo como en el grupo. De este modo, el tipo MIME "image/\*" representa a todos los archivos de imagen, ya estén almacenados en formato gif, jpeg, bmp, etc.