

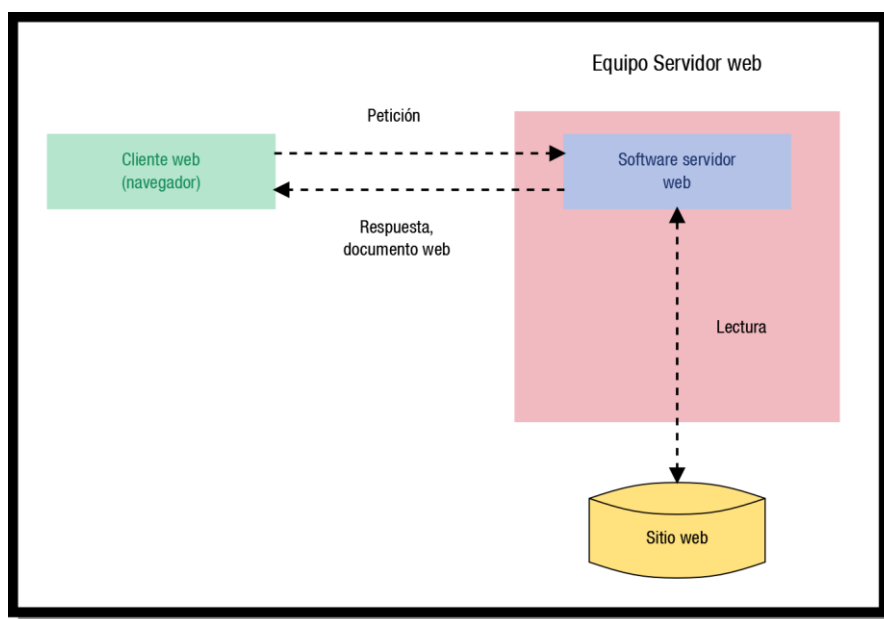
El Servicio Web y Protocolo HTTP

El servicio web es el más utilizado por los usuarios de Internet.

Hacemos uso del servicio cada vez que abrimos una página web con nuestro navegador web (Internet Explorer, Mozilla Firefox, Google Chrome, etc.).

El navegador web no es más que un **cliente web** que realiza la petición de una página web a un servidor web especificando en la petición el nombre DNS del equipo que tiene instalado el **servidor web** y la ruta y nombre de la página web solicitada.

El servidor lee en el sitio web el archivo correspondiente a la página web solicitada y se lo envía al cliente.



Conceptos previos que hay que conocer:

- ✓ **Página web:** Archivo escrito en un lenguaje de marcas que, además de texto, contiene como elementos principales hipervínculos o hiperenlaces que enlazan la página con otras partes de la misma o con otras páginas que estén en el mismo equipo o en otros equipos de la red.
- ✓ **Sitio web:** Conjunto de páginas web y archivos complementarios que se distribuyen en Internet y/o en una Intranet bajo un mismo nombre DNS. Por ejemplo, el sitio **www.mec.es** está formado por el conjunto de páginas web y archivos complementarios de los que se accede a través del nombre **www.mec.es**.

Un equipo **servidor web** tiene que tener unas características apropiadas para dar servicio a los clientes. En función del tipo de contenidos que ofrezca, del número de clientes que puedan acceder simultáneamente al servicio, del número de sitios web que sirva, etc., se deberá hacer una estimación de la memoria que deberá tener el equipo, las unidades de disco, el procesador o los procesadores, el ancho de banda de la conexión a red y otras características hardware y software.

Hay una gran variedad de software servidor web tanto libre como propietario (Microsoft IIS, Apache Web Server, Sun Java Web System Server, etc.). Las características que se deben valorar de este software son, entre otras:

- Si es software libre o propietario.
- Recursos que consume.
- Posibilidad de administrar varios sitios.
- Control de acceso a usuarios.
- Seguridad en las transmisiones.
- Integración de módulos de servidor (PHP, ASP, Perl, etc.).

URL.

Una URL es una secuencia de caracteres con un formato determinado que permite localizar un recurso en Internet o en otra red cualquiera.

Por ejemplo, la URL correspondiente a la página web principal del Ministerio de Educación y Ciencia es: **`http://www.mec.es/index.html`**.

Una URL tiene el siguiente formato:

Protocolo://usuario:contraseña@máquina:puerto/ruta_recurso

En la siguiente tabla describe cada elemento de una URL cuando el protocolo es http:

Elementos de una URL.

Elemento URL	Descripción
Protocolo://	Especifica el protocolo mediante el que se accede al recurso. Para páginas web el protocolo es http. Pueden especificarse otros protocolos como ftp .
Usuario:contraseña@	Es un elemento que se puede excluir en una URL. Se debería usar cuando en el servidor se controla el acceso al recurso mediante un nombre de usuario y una contraseña indicados en la URL. Casi nunca se utilizan y se desaconseja su utilización por seguridad. Hay otras formas de controlar el acceso y más seguras.
máquina	Es el nombre DNS o la dirección IP de la máquina donde se encuentra el servicio que proporciona el recurso que se trata de localizar. Una URL para una página web consta como mínimo del protocolo y de la máquina, como por ejemplo <code>http://www.mec.es</code> .
:puerto	Especifica el puerto del servidor con el que hay que conectar. No es obligatorio, si no se especifica se asume uno por defecto para el protocolo usado. Por ejemplo, para http se asume por defecto el puerto 80. Si hubiera que usar otro, habría que especificarlo.
/ruta_recurso	Especifica la ruta donde tiene que localizar el recurso el servidor y el nombre del recurso. Si no se especifica, el servidor asume un nombre de recurso por defecto (es habitual que sea index.html).
otros	Se especifican después de la ruta y sirven para enviar datos al servidor para que los procese. Por ejemplo, se usan cuando solicitamos una búsqueda en Google.

El Protocolo HTTP

El protocolo **HTTP (HyperText Transfer Protocol)** es un protocolo de transferencia de hipertexto que sigue el modelo cliente/servidor y establece las normas para el intercambio de la información contenida en las páginas web.

El protocolo HTTP ha sido desarrollado desde 1990 por W3C y IETF. Se han desarrollado las versiones 0.9, 1.0 y 1.1. La versión 1.1 ha sido publicada en el RFC 2616.

Las características más relevantes del protocolo de HTTP son las siguientes:

- Utiliza una **arquitectura cliente-servidor**.
- Es un protocolo del **nivel de aplicación**.
- Un **servidor** HTTP utiliza por defecto el **puerto 80**, aunque puede usar otros puertos.
- El **cliente** utiliza un **puerto mayor de 1000**.
- Para establecer una comunicación HTTP entre cliente y servidor se necesita crear previamente una **conexión TCP**.
- La comunicación entre clientes y servidores se realiza mediante **mensajes de petición y de respuesta** codificados en ASCII.
- Cada elemento de una página web (documento, imágenes, vídeos, etc.), **se transfiere independientemente** con su mensaje de petición y su mensaje de respuesta.

El contenido de una página web que se transfiere entre un servidor y un cliente está codificado en un lenguaje de marcas que describe como se ha de visualizar la página web. De los lenguajes de marcas para descripción de páginas web, sin duda es HTML el más utilizado.

Los navegadores web son clientes HTTP que interpretan el contenido de una página web escrita en HTML o en otro lenguaje para crear una representación de la página para ser visualizada por el usuario.

Cuando en un navegador web escribimos la URL de una página web:

- ✓ Se solicita al servidor DNS que resuelva el nombre de equipo servidor web o HTTP usado en la URL.
- ✓ Obtenida la IP del servidor HTTP, se establece una conexión TCP entre cliente y servidor HTTP.
- ✓ Establecida la conexión TCP, se hace la petición del documento web al servidor y éste devuelve su contenido en un mensaje de respuesta.
- ✓ Si el documento web incluye elementos adicionales como imágenes, hay un proceso de envío **petición/respuesta** por cada imagen.

Al realizar un cliente una petición a un servidor se genera un circuito abierto entre ellos.

Lo más probable es que se abra más de una conexión entre el cliente y el servidor para servir todo el contenido de la página Web. Podemos hablar de dos tipos de conexión.

- **Conexión persistente:** Se mantiene la conexión abierta entre cliente y servidor. Las conexiones persistentes dejan puertos abiertos y pueden dar problemas de seguridad ya que pueden atacar al ordenador por esos puertos abiertos.
- **Conexión no persistente:** Después de servir la página de índice se cierra la conexión (esto es lo que se hace normalmente). Si el cliente necesita algo más de esa página Web se realiza una nueva conexión por otros puertos.

Mensajes HTTP

Desde que un cliente HTTP (navegador web) establece una conexión con un servidor HTTP, para descargar una página web hasta que la descarga de forma completa, se desarrolla una **sesión HTTP**.

Normalmente, en una sesión HTTP se producen varias transacciones de mensajes de petición y respuesta entre cliente y servidor.

Por cada recurso adicional (imagen, audio, vídeo, etc.) que contiene una página web se envía un mensaje de petición del recurso desde el cliente hacia el servidor y un mensaje de respuesta desde el servidor al cliente con el recurso solicitado o con una indicación de que el recurso no se ha podido obtener.

La siguiente imagen muestra una captura de pantalla de la página web del Ministerio de Educación <http://www.educacion.gob.es/portada.html>. Sobre la captura se destacan los recursos adicionales y se representan los mensajes de petición y respuesta que enviaría y recibiría un cliente HTTP que consultase esa página web.



Los mensajes son textos codificados en ASCII y constan de cuatro campos:

- **Línea de petición o de respuesta (según sea el caso):** contiene la información principal sobre la petición o la respuesta.
- **Encabezados:** contienen información adicional sobre opciones relativas al mensaje, (una línea de encabezado por cada opción que se especifique).
- **Una línea vacía:** se ha generado con un ENTER. La consideramos como un campo aunque no es más que un separador entre las líneas de encabezados y el cuerpo del mensaje.
- **Un cuerpo de mensaje:** este campo es opcional. En los mensajes de respuesta se usa para enviar el contenido del recurso solicitado.



Mensajes de petición.

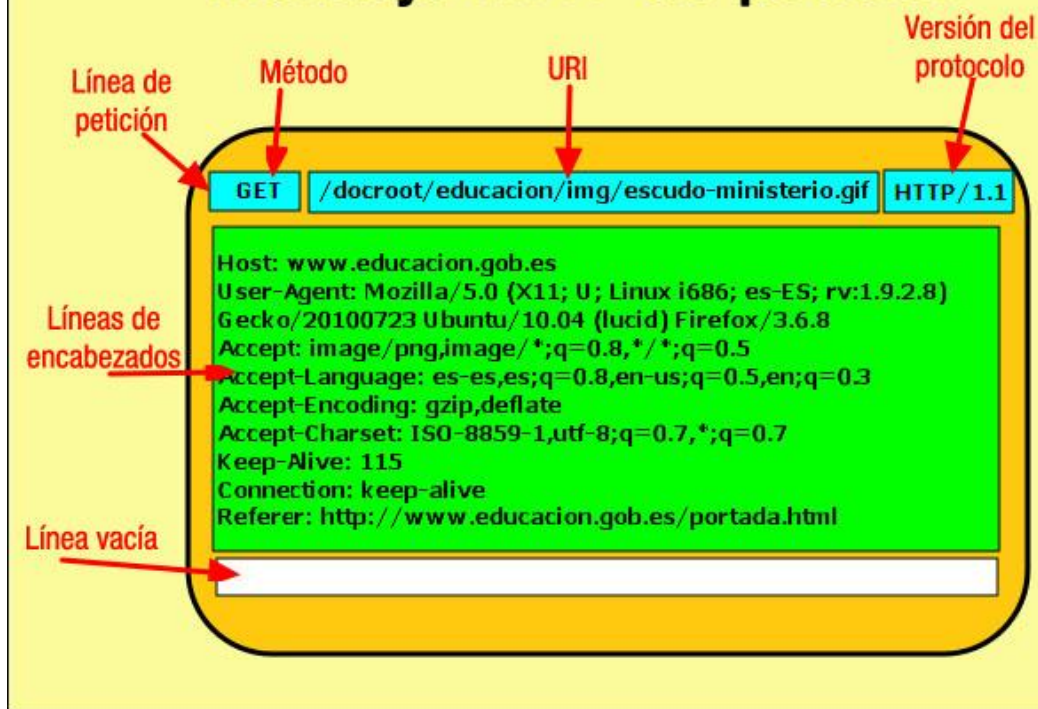
Un mensaje de petición es enviado al servidor HTTP desde el cliente.

En un mensaje de petición, la **línea de petición** contiene tres datos separados por un espacio:

- **Método de la petición.**
- **Dirección [URI](#) del recurso.**
- **Versión de HTTP.**

La siguiente imagen muestra un ejemplo de un mensaje de petición para solicitar mediante el método GET el recurso **escudo-ministerio.gif**.

Mensaje HTTP de petición



- ✓ El **método** de la petición indica la acción que se pretende realizar sobre el recurso indicado en la petición. Los métodos más utilizados son GET y POST.
- ✓ La **dirección URI** del recurso especifica la ruta del recurso dentro del directorio raíz del sitio web del que se solicita el recurso. Por ejemplo, la URI `/imag/escudo.gif` indica que el recurso solicitado es **escudo.gif** y se encuentra en el directorio **imag** bajo el directorio raíz del sitio.
- ✓ Las **líneas de encabezados** indican varias opciones relativas a la petición. Cada encabezado se representa con una línea de texto con el nombre de la opción o del encabezado, el carácter ":" y el valor asignado a la opción.
- ✓ El **cuerpo de la petición** está vacío si la petición usa el método GET y contiene información que se envía al servidor si usa el método POST.

Puedes instalar una extensión para tu navegador que te permita ver el contenido de los mensajes de petición y respuesta HTTP de las conexiones que establezcas. Para Mozilla Firefox la extensión se llama "**Live Http Headers**". Para Internet Explorer, una posible extensión es "**IEWatch**".

Métodos y encabezados de los mensajes de petición.

El **método** de una petición HTTP indica la acción que se quiere realizar sobre el recurso objeto de la petición.

- **GET:** el cliente solicita que el servidor le envíe una representación del recurso indicado en la petición. Con representación nos referimos al contenido del archivo correspondiente al recurso. Siempre que pulsamos con el ratón en un enlace de una página web a cualquier recurso o escribimos una URL en la barra de direcciones del navegador, éste envía al menos un mensaje de petición con el método GET.
- **POST:** mediante este método, el cliente envía información al recurso indicado del servidor para que sea procesada por una aplicación. Normalmente se usa este método cuando hemos rellenado un formulario de una página web y pulsamos un botón como "Enviar", "Confirmar" o cualquiera similar. En el cuerpo del mensaje se envía la información a procesar.
- **HEAD:** se solicita al servidor que envíe solamente los encabezados correspondientes a la petición del recurso que se indique. Para un mismo recurso, el servidor respondería con la misma línea de respuesta y los mismos encabezados que para una petición GET pero no enviaría el contenido del recurso. Lo usan los navegadores por ejemplo para saber si una página web que tienen almacenada en caché ha sido actualizada en el servidor.

Encabezados de peticiones HTTP

Encabezado	Significado
Host.	Equipo al que se envía la petición.
User-Agent.	Nombre y versión del cliente (navegador) y del sistema operativo.
Accept.	Tipo de contenido que acepta el navegador.
Accept-Language.	Idiomas que espera el navegador en las páginas recibidas.
Accept-Encoding.	Sistema de codificación que espera el navegador para el recurso a recibir.
Accept-Charset.	Juego o conjunto de caracteres que espera recibir el navegador.
Referer.	URL desde donde se originó la petición (URL de la página que contenía el enlace).
Cookie.	Contenido de la cookie almacenada en el equipo cliente y relativa al sitio web al que se hace la petición.

Mensajes de respuesta.

Los mensajes HTTP de respuesta son enviados por el servidor HTTP a clientes HTTP en respuesta a una petición. En un mensaje de respuesta hay:

- ✓ **Una línea de petición.**
- ✓ **Varias líneas de encabezados**
- ✓ **Una línea vacía**
- ✓ **Un cuerpo del mensaje.** En el cuerpo del mensaje, el servidor envía el contenido del recurso que se le ha solicitado (cuando se solicita su envío). Por ejemplo, puede contener el código HTML de una página web.

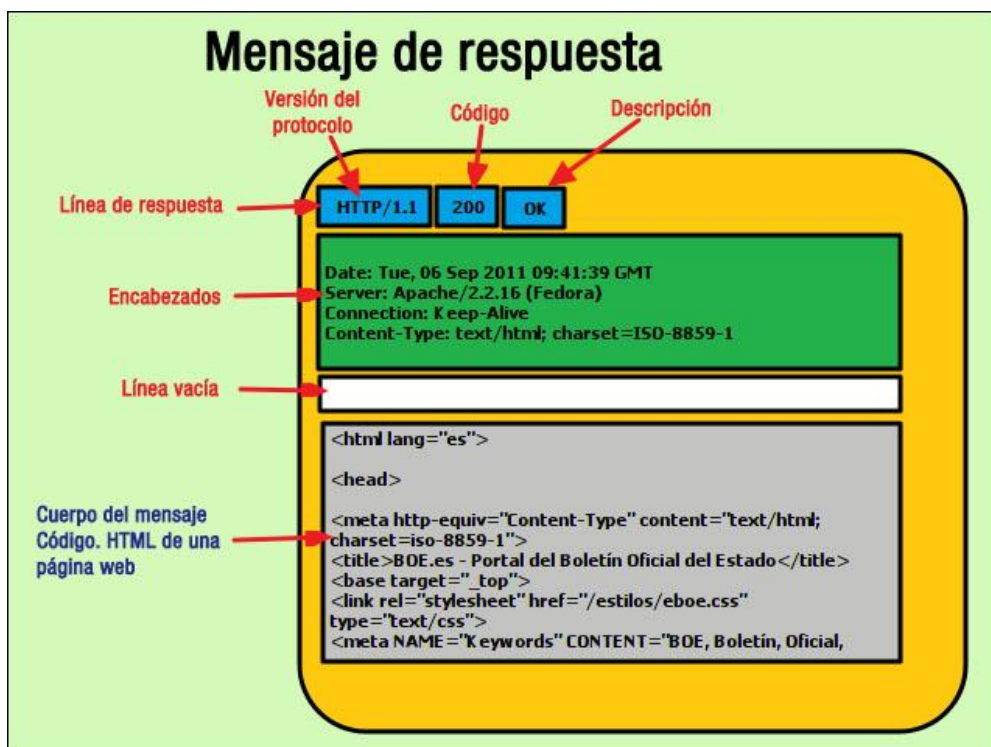
En un mensaje de respuesta, la **línea de respuesta** (también llamada **línea de estado**) contiene tres datos:

- ✓ **Versión del protocolo.**
- ✓ **Código de respuesta o código de estado.**
- ✓ **Descripción de respuesta o descripción de estado.**

El **código de respuesta** es un valor numérico que representa como ha sido recibida y procesada la petición a la que se está respondiendo. Sirve para indicar si una petición se ha recibido y atendido correctamente, o si se ha producido cualquier problema o circunstancia que haya alterado una normal recepción y atención de la petición.

La **descripción** es una frase corta que describe lo que se está indicando con el código de respuesta (a cada código corresponde una descripción).

La siguiente imagen muestra cada uno de los elementos que forman parte de un mensaje de respuesta HTTP.



Encabezados de respuestas HTTP

Encabezado	Significado
Date.	Fecha y hora en que ha comenzado el envío del recurso solicitado.
Server.	Nombre del software servidor.
Content-Length.	Tamaño en bytes del cuerpo del mensaje.
Content-Language.	Código del idioma en que está escrito el recurso que se envía en el cuerpo del mensaje.
Content-Encoding.	Sistema de codificación que se utiliza en la representación del recurso dentro del cuerpo del mensaje.
Connection.	Sirve para indicar si el servidor mantiene la conexión TCP para las siguientes transacciones HTTP desde el cliente. Si tiene el valor keep-alive mantiene la conexión y si tiene el valor close la cierra, en cuyo caso, si el cliente tiene que enviar una nueva petición HTTP tendrá que solicitar previamente una nueva conexión TCP.

Servidores web

A la hora de decidirnos por un servidor web o HTTP, podemos optar entre varios servidores disponibles. En la siguiente tabla se listan y describen los servidores web más utilizados.

Servidores web

Servidor	Descripción
Apache.	Es el más usado en Internet. Desarrollado por Apache Software Foundation. Es software libre, gratuito y multiplataforma (Unix, Linux, Windows. Netware).
Microsoft IIS.	Es software propietario y sólo se puede instalar en sistemas Windows.
Sun Java System Web Server.	Está desarrollado por Sun y se usa principalmente en sistemas Sun. Actualmente es software libre, gratuito y multiplataforma.
Tomcat.	Al igual que Apache está desarrollado por Apache Software Foundation. Es software libre, gratuito y multiplataforma. Usa una tecnología distinta a la de Apache. Permite trabajar con algunas páginas dinámicas con las que no puede trabajar Apache. Puede trabajar como un servidor de aplicaciones.
Ngnix.	Es un servidor que consume muy pocos recursos. Se puede instalar en sistemas Unix/Linux y Windows. Es software libre y gratuito.
Lighhttp.	Es también un servidor que consume muy pocos recursos y muy fácil de configurar con su herramienta gráfica de configuración. Es software libre, gratuito y multiplataforma.
Cherokee.	Su principal objetivo es ser rápido y muy funcional. Es software libre, gratuito y multiplataforma.

CONSTRUCCIÓN DE LA TRAMA HTTP

Nivel de Aplicación

Quien inicia el proceso de comunicación es el cliente poniendo en el navegador el nombre de la página que quiere consultar.

Estamos trabajando a nivel de aplicación: comando GET para indicarle al servidor que la consulta es para obtener una página web y la Web que quiere consultar.

Datos	GET	http://192.168.5.100
-------	-----	----------------------

Nivel de Transporte

Ahora se realiza una solicitud de conexión con el servidor.

Se construye la unidad de transmisión correspondiente a este nivel que es el **segmento**, encapsulando en el campo datos del segmento los datos del nivel de aplicación.

Segmento	Cabecera			
	Puerto Destino	80		
	Puerto Origen	1001		
	Tipo Protocolo	http		
	<table border="1"><tr><td>Datos</td><td>GET</td><td>http://192.168.5.100</td></tr></table>		Datos	GET
Datos	GET	http://192.168.5.100		
Control Errores				

Este es un **segmento TCP**, es decir, fiable y por tanto hay que responder a esta solicitud.

Para construir este segmento es necesario que se reserve un puerto, en el ejemplo vamos a suponer que reserva el puerto 1001.

Nivel de Internet

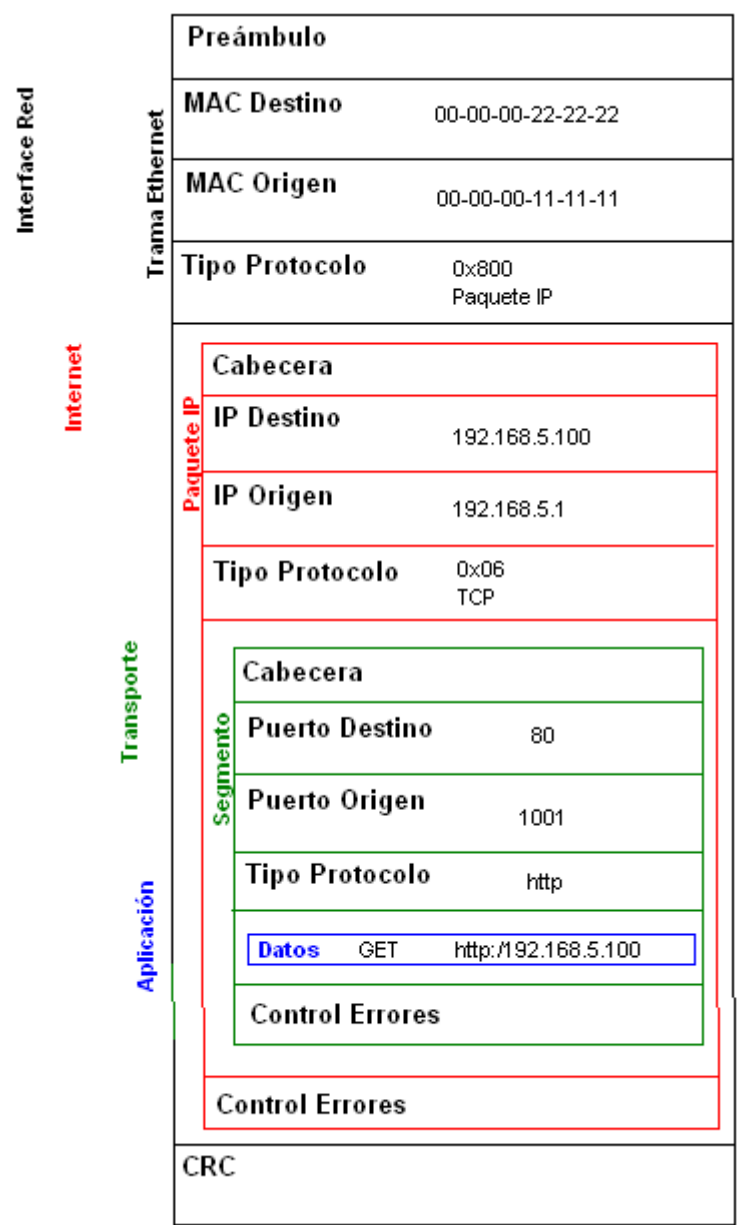
Construye su unidad de transmisión: el **paquete IP**.

Paquete IP	Cabecera	
	IP Destino	192.168.5.100
	IP Origen	192.168.5.1
	Tipo Protocolo	0x06 TCP
Segmento	Cabecera	
	Puerto Destino	80
	Puerto Origen	1001
	Tipo Protocolo	http
	Datos	GET http://192.168.5.100
	Control Errores	
Control Errores		

La **IP Destino** la toma del dato del explorador o la ha obtenido mediante DNS, y la **IP Origen** del registro del sistema.

Nivel de Interface de Red (Enlace)

Construye su unidad de transmisión: la **trama**.



La **MAC Origen** es conocida. La **MAC destino** tiene que averiguarla, para ello:

- Consulta su tabla ARP (*arp - a*).

En caso de que no esté la MAC de esa IP,

- Lanza una trama ARP para averiguarla.

Nivel de Interface de Red (Físico)

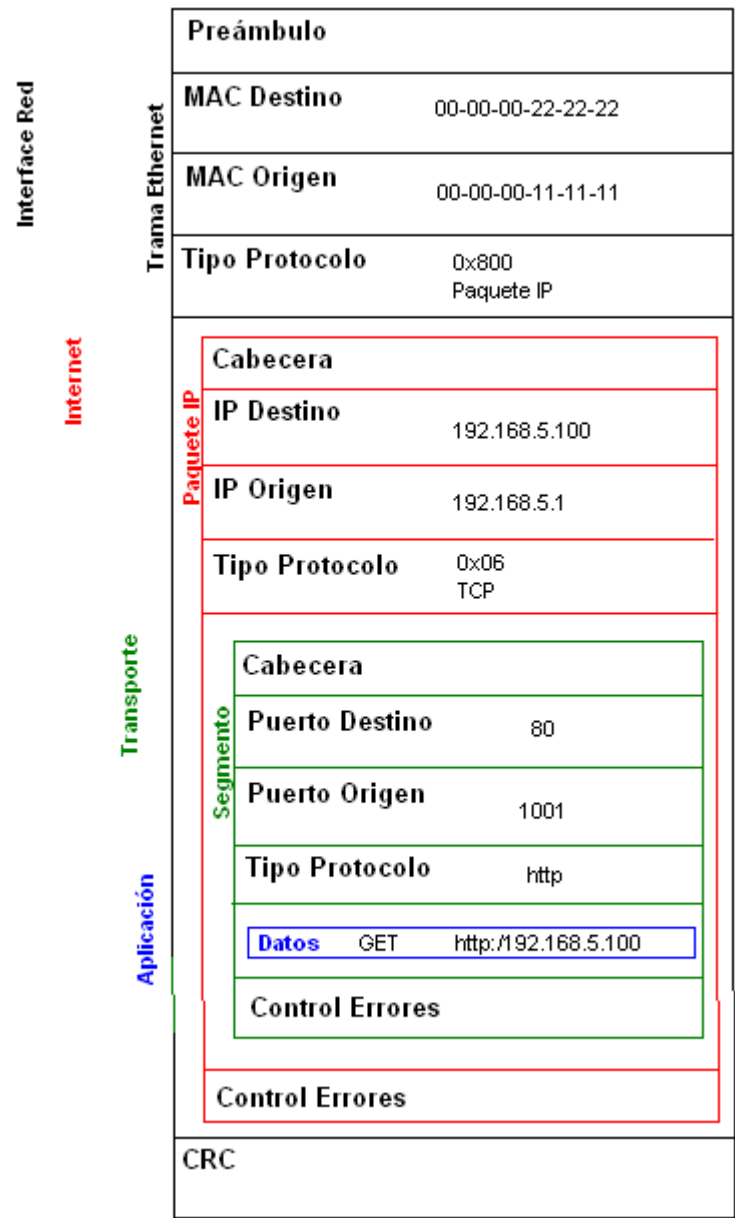
El proceso de encapsulación ha terminado. La tarjeta de red es la encargada de traducir los datos de la trama en *0* y *1* (niveles de tensión) que viajarán por el medio de transmisión.



Los datos llegan a la tarjeta de red del servidor como *0* y *1*. La tarjeta de red realiza el proceso inverso, a partir de los niveles de tensión reconstruye la trama.

DESENCAPSULACIÓN

Nivel de Interface de Red (Enlace)



El servidor de la Trama que recibe primero analiza la MAC Destino, se pregunta si la trama es para él.

Si la trama no es para él la elimina sin más.

Si la trama es para él, apunta en un buffer la MAC Origen por si la necesita posteriormente.

En el campo Tipo Protocolo de la trama ve que el contenido de la misma es un paquete IP.

Desencapsula la trama para obtener el paquete IP y poder pasar al siguiente nivel de la arquitectura TCP/IP.

Nivel de Internet

Paquete IP	Cabecera	
	IP Destino	192.168.5.100
	IP Origen	192.168.5.1
	Tipo Protocolo	0x06 TCP
Segmento	Cabecera	
	Puerto Destino	80
	Puerto Origen	1001
	Tipo Protocolo	http
	Datos	GET http://192.168.5.100
	Control Errores	
Control Errores		

Analiza el campo IP Destino y comprueba que la trama es también para él a este nivel de la arquitectura, luego tiene que seguir analizando la trama.

Con la IP Origen ya puede apuntar en su tabla ARP la entrada obtenida:

192.168.5.1 00-00-00-11-11-11

En el Tipo de Protocolo observa que tiene un segmento TCP, luego está obligado a contestar.

Continúa el proceso de desencapsulamiento.

Nivel de Transporte

Segmento	Cabecera	
	Puerto Destino	80
	Puerto Origen	1001
	Tipo Protocolo	http
	Datos	GET http://192.168.5.100
	Control Errores	

El **tipo de Protocolo** le indica que la trama está relacionada con el servicio Web, y que la petición llega por el puerto 80.

Apunta el puerto al que tiene que responder.

Sigue con el proceso de desencapsulamiento.

Nivel de Aplicación

Datos	GET	http://192.168.5.100
-------	-----	----------------------

El método GET indica que se está haciendo una petición de página Web.

Ahora el servidor debería construir la trama de respuesta con los datos que ha ido guardando al analizar la trama.

En este caso en el campo Datos del nivel de aplicación será el código de la página Web que tiene que servir para que luego el Explorer la interprete y presente en el navegador.

Así la trama que saldrá del servidor será la siguiente:

