



UT5. Gestión de servidores web (IIS)

Javier Rojas



1. Introducción

- HTTP(Hyper Text Transfer Protocol)es un protocolo de capa de aplicación que facilita a los usuarios, de una forma sencilla e intuitiva, el acceso a la información hipermedia remota de sistemas conectados a una red TCP/IP.
- El modelo cliente/servidor y el protocolo HTTP son la base de la WWW (World Wide Web o simplemente Web)

2. WWW

- La WWW(World Wide Web)es un servicio de distribución de información que permite acceder a millones de recursos electrónicos (documentos, imágenes, audio, vídeo,...) y aplicaciones distribuidos en servidores por todo Internet e identificados y localizados por direcciones (URIs o URLs).
- Estos recursos están conectados entre sí a través de hiperenlaces (o hipervínculos) lo que permite "navegar" de unos a otros fácilmente.

3. W3C y estándares Web

- La WWW fue desarrollada por el CERN (Centro Europeo de Investigación Nuclear) en el año 1989 y actualmente su desarrollo está controlado por el W3C (World Wide Web Consortium) (<http://www.w3.org/>), una comunidad internacional que desarrolla estándares web, por ejemplo XHTML, CSS y XML, que aseguran el crecimiento de la Web a largo plazo.

Actividad

- Busca información en Internet sobre posicionamiento en buscadores o posicionamiento web (SEO, Search Engine Optimization). Investiga sobre el Pagerank de Google ¿cómo puedes saber el Pagerank de una web?
- Consulta la web www.w3.org del W3C y observa los estándares que desarrolla. Pincha en el hiperenlace W3C A to Z para tener una visión general de todos los estándares de la Web.

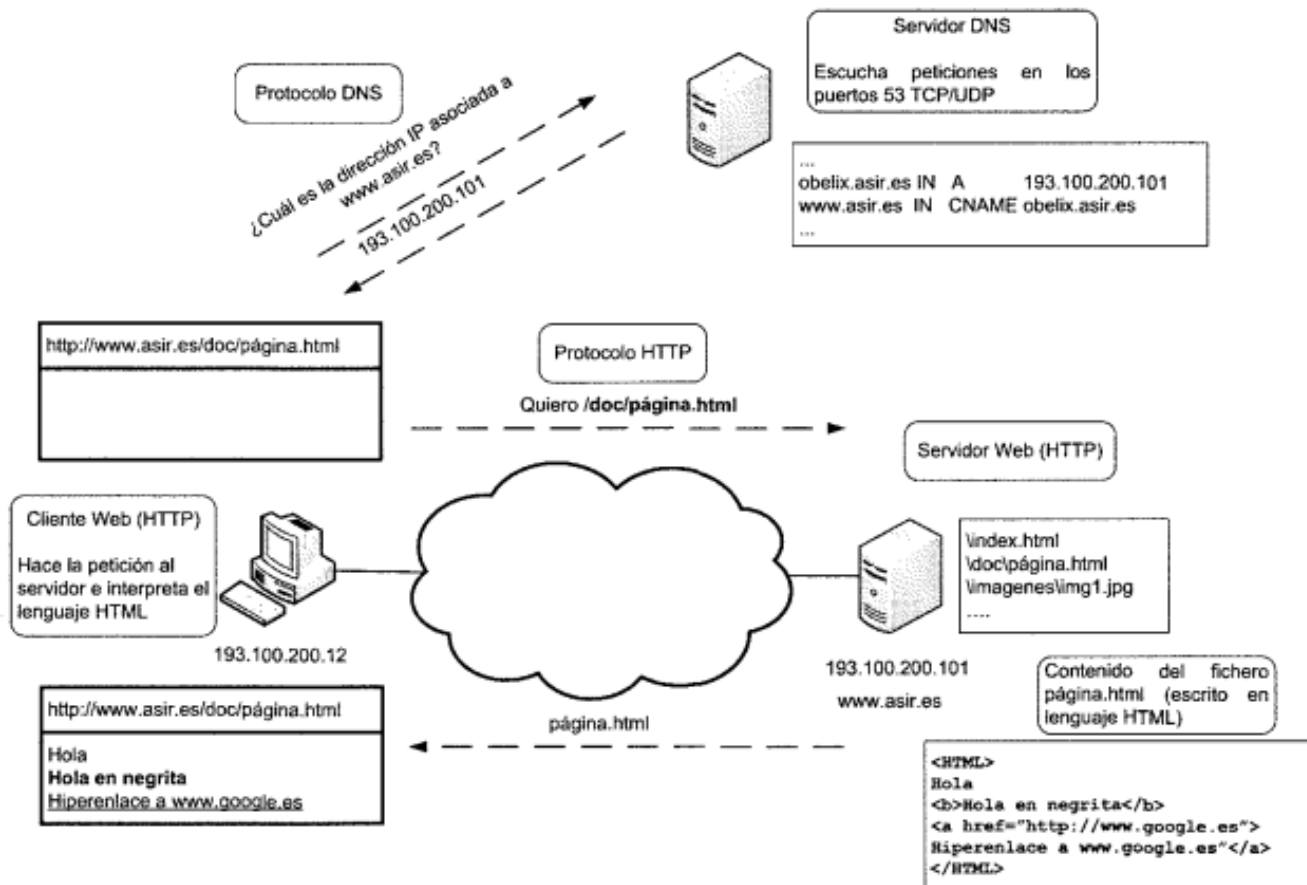
4. Componentes y funcionamiento

- El servicio que ofrece la Web se basa en el modelo cliente/servidor y está formado por los siguientes componentes.
 - Recursos. Documentos, vídeos, imágenes, audio, aplicaciones, buzones de correo, etc., accesibles a través de servidores web y conectados por hiperenlaces.
 - Nombres y direcciones (URIs y URLs). Sistema de nombres basado en cadenas de caracteres que identifican y localizan inequívocamente a los recursos en la Web.
 - Clientes web (clientes HTTP o navegadores). Permiten a los usuarios acceder a los recursos disponibles en servidores web. Establecen conexiones con los servidores web, dialogan con ellos e interpretan la información que obtienen mostrándosela a los usuarios.

4. Componentes y funcionamiento

- Servidores web (o servidores HTTP). Atienden las peticiones de los clientes y les envían los recursos solicitados.
- Proxies web (o proxies HTTP). Programas intermediarios entre clientes y servidores web. Pueden actuar como cortafuegos y/ o almacenar datos en cache para aumentar el rendimiento.
- Protocolo HTTP. Conjunto de normas y reglas en base a las cuales "dialogan" los clientes, los servidores web y los proxies. Usa TCP como protocolo de transporte.
- Tecnologías web. Utilizadas para desarrollar aplicaciones basadas en la Web (XHTML, CSS, XML, Ajax, XQuey, XPath, RDF, etc.)

4. Componentes y funcionamiento



5. Nombres y direcciones (URIs y URLs)

- La Web se puede ver como un conjunto de recursos distribuidos y conectados entre sí. Para poder localizar los recursos y acceder a ellos se utilizan cadenas de caracteres:
- por ejemplo, <http://www.asir.net/pagina.html>
- que los identifican inequívocamente. Estas cadenas de caracteres se denominan identificadores uniformes de recursos (URI, Uniform Resource Identifier) y permiten acceder a los recursos disponibles usando una gran variedad de sistemas de denominación y métodos de acceso, llamados esquemas, como HTTP, FTP, etc.
- Los localizadores universales de recursos (URL, Universal Resource Locator) son tipos de URIs.

Actividad

- Existe cierta confusión en la comunidad web sobre la relación y de los términos URI, URL y URN (Uniform Resource Name). Se debe a la incompatibilidad de dos visiones diferentes de la estructura de una URI, denominados punto de vista clásico y contemporáneo. Puedes ampliar información, profundizar sobre los formatos de URIs, URLs y URNs e intentar aclarar la confusión existente consultando la web
- <http://www.w3.org/Addressing/>
- Haz un esquema de media página aclarando las diferencias y dando ejemplos

Actividad

- <http://193.168.200.101/pagina.html>
 - http. Esquema o protocolo que se utiliza para acceder al recurso.
 - 192.168.200.101. Dirección IP de la máquina donde está el recurso.
 - pagina.html. Ruta del recurso solicitado relativa al directorio raíz del servidor web.
- <http://aula.asir.es:8080/datos/practica1.pdf>
- <http://obelix.asir.es/buscarLibros.php?id=2&tema=Historia>

6. Páginas web, sitios web y aplicaciones web

- Una página web es un documento hipermedia o conjunto de información electrónica relacionada (texto, audio, imágenes, vídeo, etc.) que normalmente contiene hiperenlaces a otras páginas web o recursos. Las páginas web están escritas en lenguajes que son interpretados y/ o ejecutados por los navegadores (XHTML, CSS, Java Script, Flash, ...). Su contenido puede ser estático (almacenado en un servidor web) o dinámico (se genera en el servidor web al ejecutar un conjunto de instrucciones de un determinado lenguaje).

6. Páginas web, sitios web y aplicaciones web

- Un sitio web es un conjunto de páginas web relacionadas y accesibles a partir de un mismo nombre de dominio DNS. El conjunto de sitios web de Internet constituyen la WWW. Los sitios web se pueden clasificar según múltiple criterios, uno de los más empleados es el tema o contenidos que ofrecen (sitios de redes sociales, sitios de periódicos, sitios de buscadores, etc.).
- Una aplicación web es aquella donde el usuario interactúa con un navegador que accede a los servicios y recursos que ofrece un servidor web (por ejemplo, un buscador, una tienda electrónica, un cliente de correo web, ...).

7. Servidores web

- Los servidores web, también llamados servidores HTTP, son programas que atienden peticiones HTTP, procesan e interpretan código escrito en diferentes lenguajes y envían a los clientes los recursos solicitados.
- Los recursos pueden estar localizados en el mismo equipo donde se ejecuta el servidor o en otros equipos de la red a los que el servidor puede acceder usando protocolos adicionales. El servidor puede enviar tanto contenido estático (archivos en diferentes formatos) como dinámico (el resultado de ejecutar programas).
- Ofrecen múltiples opciones de configuración y suelen tener una arquitectura modular que permite ampliar o quitar funcionalidades fácilmente.

7. Servidores web

- Por defecto escuchan peticiones
- HTTP en el puerto 80 /TCP.
- Existen múltiples servidores web tanto para sistemas libres como para sistemas propietarios.
- Algunos de los más utilizados son:
 - Apache
 - IIS
 - Nginx
 - Lighttpd

Actividad

- Netcraft: es una compañía que, entre otros servicios, ofrece análisis y comparativas de la Web. Visita su página web.
- <http://www.netcraft.com/>
- En la parte derecha de la página inicial de Netcraft hay un formulario "What's that site running?" que permite consultar información sobre el servidor web en el que se ejecuta o aloja un sitio web determinado. Consulta 5 de los sitios que visitas habitualmente.
- Busca estadísticas de uso de servidores web (web survey).

8. Clientes web (navegadores)

- Los clientes web o navegadores son programas con los que interactúa el usuario y que le permiten, entre otras funciones, introducir URIs (o URLs) para acceder a recursos disponibles en la red. pueden actuar como clientes de diferentes protocolos aunque su función principal es ejercer como clientes HTTP.
- Reciben recursos de los servidores web, los procesan y muestran los resultados al usuario, permitiéndole interactuar si es necesario. Si el recurso que recibe el navegador no puede ser interpretado por él, puede redirigirlo a una aplicación externa capaz de gestionarlo o preguntar al usuario qué quiere hacer.

8. Clientes web (navegadores)

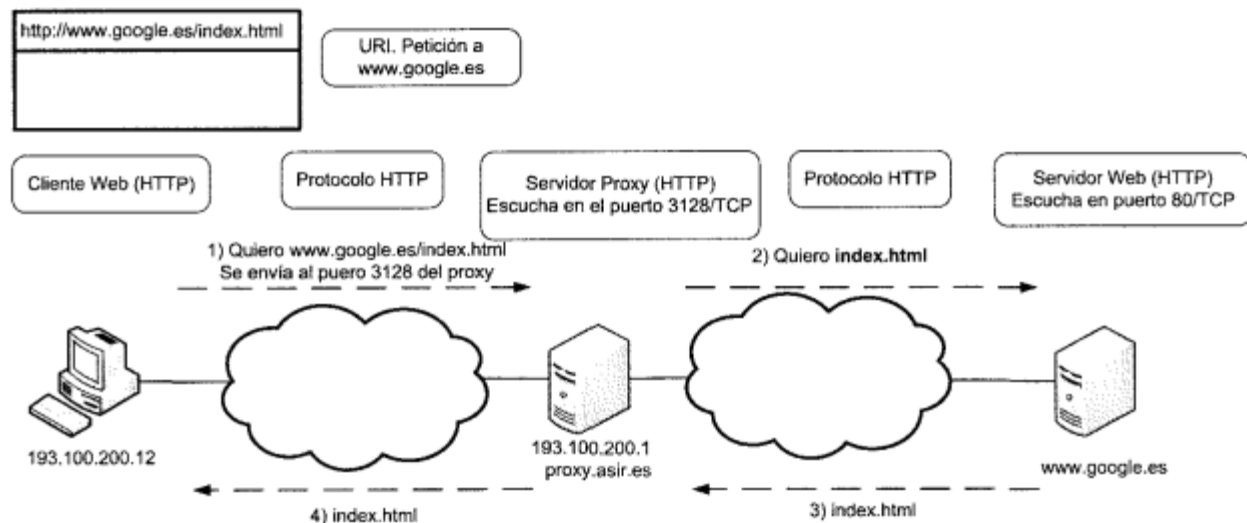
- Mantienen una memoria cache en la que almacenan durante un tiempo las direcciones a las que han accedido (historial), los recursos procesados, las contraseñas introducidas por el usuario en las aplicaciones, etc.
- Los navegadores ofrecen múltiples opciones de configuración y personalización. Además, permiten ampliar su funcionalidad con la instalación de plantillas, idiomas, extensiones y complementos.
- Aunque la WW3C crea estándares para tecnologías web, los navegadores, en determinados aspectos, no cumplen completamente los estándares. Esto supone un problema para los desarrolladores de aplicaciones web, que tienen que adaptarlas para que se muestren y funcionen correctamente en varios navegadores, incluso entre diferentes versiones de un mismo navegador. También es un problema para los usuarios porque accediendo al mismo sitio web con distintos navegadores pueden obtener resultados diferentes.

9. Proxies web

- En el ámbito de las redes, el término proxy hace referencia a un programa que actúa como intermediario entre otros. Un proxy web o proxy HTTP hace de intermediario entre un cliente web y un servidor web. Según el comportamiento y la función que realizan pueden ser proxies directos o proxies inversos:

9.1 Proxy directo

- Proxy directo (forward proxy). Recibe la petición iniciada por un cliente web y se la traslada al servidor web. La solicitud del cliente es hacia el servidor web (la URI que utiliza es la del servidor) no hacía el proxy, este solo hace de intermediario o representante del cliente.
- Usados habitualmente para optimizar y controlar el acceso a redes externas, como Internet, de los clientes de una organización o empresa.



9.2 Proxy inverso

- Proxy inverso (reverse proxy). Igualmente reciben la petición de un cliente web y la reenvían a uno o varios servidores web. En este caso, la solicitud del cliente (la URI que utiliza es la del proxy) es hacia el proxy (para los clientes es un servidor web normal).
- Usados habitualmente para ofrecer acceso a servidores web que están detrás de un cortafuegos y no son accesibles directamente, para balancear la carga entre varios servidores web, para aumentar los acceso ofreciendo almacenamiento en cache, etc.
- Observa que un proxy inverso puede brindar acceso a múltiples servidores web en la misma URI



9.2 Proxy inverso

- Copiad dibujo de la pizarra
- Existen proxies específicos para otros servicios como por ejemplo FTP y SMTP. Hacemos referencia en el libro a los proxies web porque son los más utilizados.

10. Protocolo HTTP

- HTTP (HyperText Transfer Protocol) es defacto el protocolo de comunicación en la Web. Define las reglas que utilizan los componentes software (clientes, servidores y proxies) para comunicarse.
- Es un protocolo sin estado que utiliza TCP como protocolo de transporte y determina los tipos de peticiones que los clientes pueden enviar, así como el formato y estructura de las respuestas.
- También define una estructura de metadatos, en forma de cabeceras que se envían tanto en las peticiones como en las respuestas.
- Ha pasado por varias versiones HTTP /0.9 (Obsoleta), HTTP /1.0, HTTP /1.1 (versión actual) y HTTP /1.2 (experimental).

10.1. Funcionamiento básico

- El usuario introduce una URI (o URL) en la barra de direcciones del navegador o hace clic sobre un hipervínculo.
- El navegador analiza la URL y establece una conexión TCP con el servidor web (si se ha utilizado un nombre DNS previamente invoca al resolver para que le resuelva el nombre) (si no se indica el número de puerto se conectará al puerto 80 del servidor).
- Cuando se ha establecido la conexión TCP, el navegador envía un mensaje HTTP de petición que depende de la URI (o URL).
- El servidor envía un mensaje de respuesta que depende de la petición enviada y del estado del servidor.
- Se cierra la conexión TCP.

Actividad

- Inicia sesión en un XP con un usuario con privilegios de administrador.
- Inicia una captura con Wíreshark, abre el navegador web e introduce la dirección de la web que deseas. Observa el tráfico capturado, analiza el inicio de conexión TCP y los mensajes de petición y respuesta intercambiados entre el cliente y el servidor. Puedes situarte sobre un mensaje
- HTTP hacer clic con el botón derecho del ratón y seleccionar Follow TCP Stream para analizar el intercambio de mensajes HTTP.
- Adjunta una captura de un mensaje de petición y otro de respuesta, analiza los campos que devuelve.

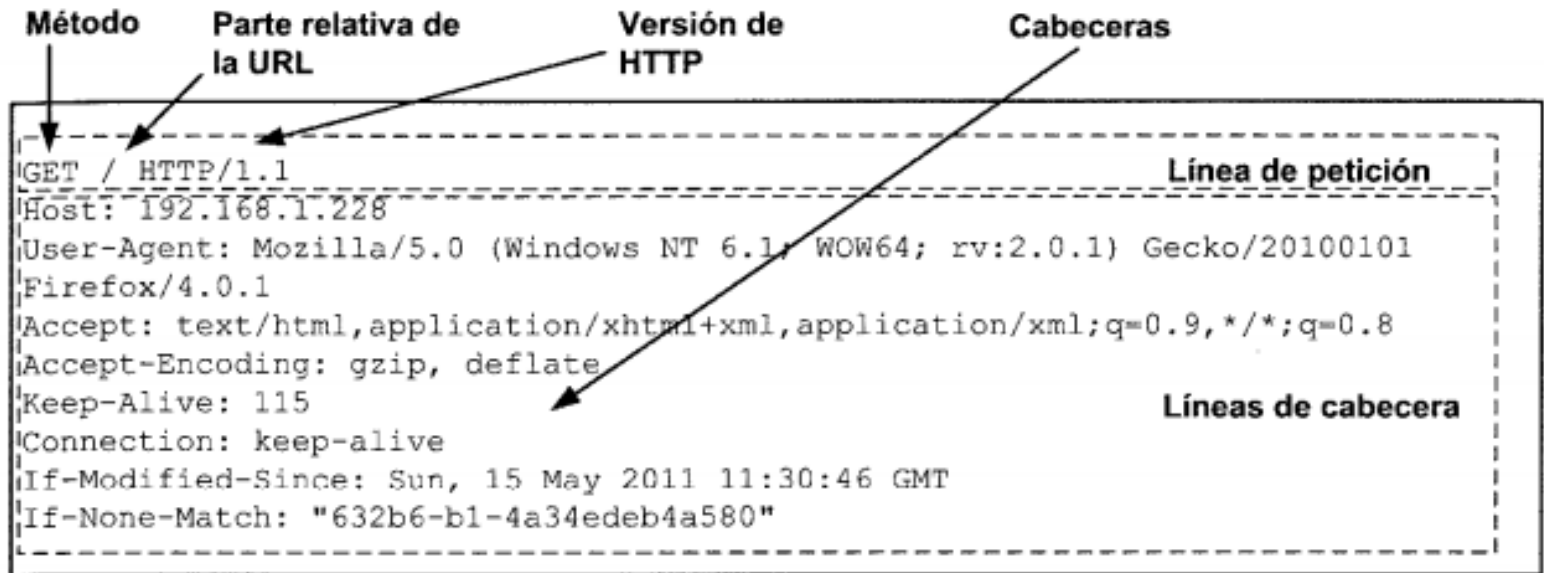
10.2. Mensajes HTTP

- Los mensajes que utiliza HTTP se componen de líneas escritas en texto plano (ASCII) que contienen las órdenes y parámetros con la sintaxis definida por el protocolo. Pueden ser de dos tipos: mensajes de petición y mensajes de respuesta.
- En los siguientes apartados se describen en primer lugar la estructura de los mensajes de petición y respuesta HTTP y posteriormente, se explican las características y la utilidad de los diferentes elementos que componen estos mensajes.

10.2.1. Mensajes de petición

- Los mensajes de petición están formados por tres partes:
- Línea inicial de petición. Incluye:
 - Método utilizado (GET, POST, ...) (se explica en apartados posteriores).
 - La parte relativa al servidor de la URL o la URL completa si la conexión se establece con un servidor proxy.
 - Versión del protocolo utilizada (opcional). Los clientes y servidores actuales usan HTTP /1.1.
- • Línea/s de cabecera
 - Conjunto de pares nombre/valor denominados cabeceras (se explican en apartados posteriores) que determinan cómo será procesada la petición por parte del servidor. Por ejemplo, la cabecera Accept: text/html indica que el navegador es capaz de procesar código HTML.
 - Si no hay cabeceras se envía un O.
 - Cada cabecera se muestra en una línea, es decir las cabeceras se terminan con CR-LF.
 - Detrás de la última cabecera se envía una línea en blanco.
- Cuerpo del mensaje (opcional). Contiene parámetros o ficheros a enviar al servidor.

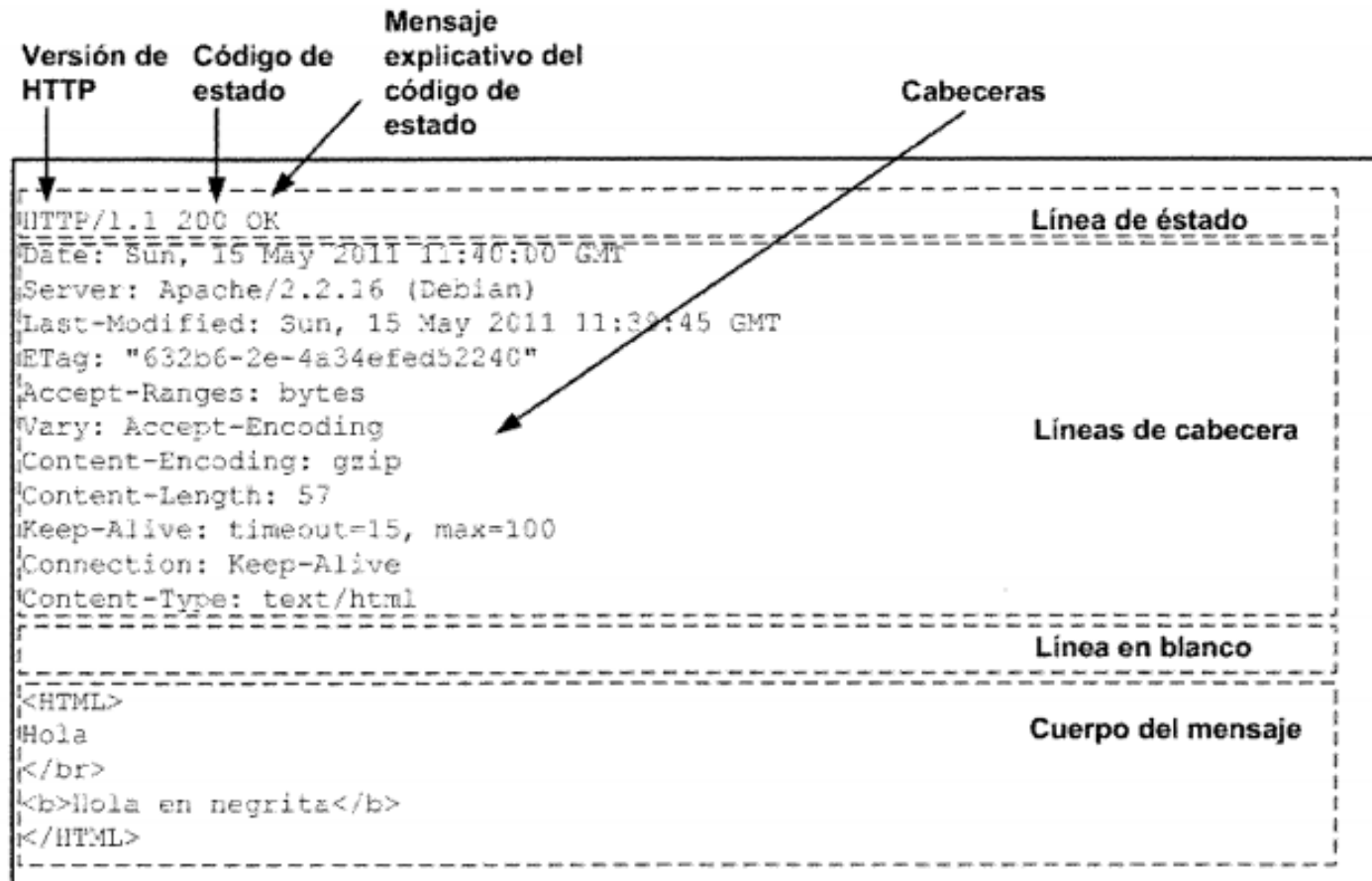
10.2. Mensajes HTTP



10.2.2. Mensajes de respuesta

- Los mensajes de respuesta están formados por tres partes:
- Línea inicial de respuesta (línea de estado)
 - Versión HTTP utilizada.
 - Código de estado o código de error que informa al cliente de cómo ha sido procesada la petición. Por ejemplo, el código 200 indica que la petición se ha procesado correctamente que el recurso correspondiente se envía al cliente. Texto explicativo del código de estado.
- Líneas de cabecera
 - Conjunto de pares nombre/valor denominados cabeceras (se explican en apartados posteriores) que describen los datos y la forma en que son enviados al cliente. Por ejemplo, la cabecera Content-Type:text/html indica que se envía código HTML que deberá interpretar el navegador.
 - Si no hay cabeceras se envía un 0.
 - Cada cabecera se muestra en una línea, es decir, la cabeceras se terminan con CR-LF.
 - Detrás de la última cabecera se envía una línea en blanco.
- Cuerpo del mensaje (opcional). Queda determinado por el tipo de recursos solicitado.

10.2.2. Mensajes de respuesta



10.2.2. Mensajes de respuesta

- Las imágenes no se envían directamente en el código HTML. Para hacer referencia a una imagen se utiliza la etiqueta `img`. Por ejemplo:
- `.`
- Cuando un navegador recibe una etiqueta `img` comprueba si tiene una copia válida en su cache y si no envía otra petición HTTP al servidor. Por lo tanto, una misma página web puede generar múltiples peticiones HTTP, tantas como imágenes tenga.
- Este comportamiento puede ocurrir con muchos otros elementos usados en lenguajes para crear páginas y aplicaciones web. Por lo tanto, cuando el navegador recibe un recurso del servidor puede darse el caso de que internamente (sin intervención del usuario) se realicen múltiples peticiones HTTP a otros recursos.



10.3. Métodos de petición

- Los métodos de petición especifican la operación que quiere realizar el cliente en el servidor. La versión HTTP 1.1 contempla siete métodos de petición que describimos brevemente a continuación:



5.10.3. Métodos de petición

- Método GET
- Método POST
- Método OPTIONS
- Método HEAD
- Método PUT
- Método DELETE
- Método TRACE



Actividad

- Completa la información: para qué se usan los métodos anteriores.
- Vuelve a realizar la captura de Wireshark para mostrar el mensaje de petición y de respuesta tal y como te los he enseñado
- Realiza pruebas para conseguir una captura de una petición post y otra get: levanta un servidor wamp y crea un formulario para analizar su tráfico.



10.4. Cabeceras

Las cabeceras son pares de nombre/valor que se pueden incluir en los mensajes de petición y respuesta HTTP. Definen información (metadatos) sobre los datos que se intercambian los clientes y servidores, sobre los propios clientes y servidores, y sobre la propia transferencia de información.

Tienen el mismo formato que las cabeceras de los protocolos usados en los servicios de correo electrónico y noticias.

Existen una gran número de tipos de cabeceras definidas por HTTP y se pueden clasificar como se muestra a continuación.



10.4.1 Generales

- Definen información que puede ser utilizada tanto por clientes como por servidores, ya que se aplican a una sesión completa de comunicación.
- Proporcionan información sobre:
 - o Control de cache (por ejemplo Cache-Control).
 - o Fechas (por ejemplo Date).
 - o Codificación de la transferencia (por ejemplo, Transfer-Encoding).
 - o etc.

10.4.2 De petición (o de cliente)

- Empleadas por los clientes para enviar información al servidor.
- Proporcionan información sobre:
 - o El propio navegador (por ejemplo User-Agent).
 - o Nombre y puerto del servidor al que se dirige la petición (Host).
 - o Tipos MIME, compresión, mapas de caracteres, idiomas que está dispuesto a aceptar el navegador, etc. (Por ejemplo Accept).
 - o Cookies (por ejemplo Cookies).

10.4.3 De respuesta (o de servidor)

- Empleadas por el servidor para enviar información añadida al cliente.
- Proporcionan información sobre:
 - o La edad de la respuesta (por ejemplo Age).
 - o Si un recurso no está disponible, fecha en la que se espera que esté disponible (por ejemplo Retry-After).
 - o El tipo servidor (por ejemplo Server).
 - o Si servidor necesita autorización para acceder al recurso pedido (por ejemplo WWW-Authenticate).



10.4.4 De entidad

- Información relacionada directamente con el recurso que se le va a proporcionar al cliente.
- Proporcionan información sobre:
 - o Codificación (por ejemplo Content-Encoding).
 - o Idioma (por ejemplo Content-Language).
 - o Longitud (por ejemplo Content-Length).
 - o Tipo MIME de los recursos (por ejemplo Content-Type).



10.5. Códigos de estado y error

Códigos que envían los servidores en las respuestas HTTP y que informan al cliente de cómo ha sido procesada la petición. Se acompañan de un texto explicativo.

Son números de tres cifras que se clasifican en 5 grupos en función del primer dígito:

10.5. Códigos de estado y error

100 - 199 (Informativo, Informational)

- Indican que el servidor ha recibido la petición pero no ha finalizado de procesarla.

200 - 299 (Éxito, Successful)

- Usados cuando la petición ha sido procesada satisfactoriamente.
- Ejemplo: 200 "OK".

300 - 399 (Redirección, Redirection)

- Indican que la petición ha sido procesada y redirigida a otra localización.

10.5. Códigos de estado y error

400 - 499 (Errores del cliente, Client Error)

- El servidor indica que hay un algún error en la petición del cliente o que no se puede conceder.

Ejemplos:

400 "Bad request". Error de sintaxis en la petición.

401 "Unauthorized". Un usuario anónimo no está autorizado para acceder al recurso solicitado.

403 "Forbidden". El servidor no acepta la petición.

404 "Not found". El recurso solicitado no está disponible en el servidor,

10.5. Códigos de estado y error

500-599 (Errores en el servidor, Server Error)

- El servidor no puede atender una petición porque ha existido algún problema.

- Ejemplos:

500 "Internal Server Error". Error interno en el servidor, por ejemplo por un fallo de configuración.

503 "Service Unavailable". El servidor no puede responder en ese momento, por ejemplo porque está sobrecargado.

5.10.6. Almacenamiento en cache (caching)

Se definen cabeceras (Cache-Control, Last-Modified, Expires, Age, Etag, If-Match, IF-ModifiedSinze, ...) que permiten controlar qué se almacena en cache, cuánto tiempo, qué no se puede almacenar (¿por qué almacenar el resultado de un partido de baloncesto que cambia constantemente?), si la información se puede "cachear" en un proxy o no, si hay que actualizar algo que está en cache porque se ha modificado, etc.

10.7. Redirecciones

HTTP permite a los servidores y a los proxies redirigir las peticiones a otras localizaciones.

Como ya sabemos, el servidor le indica la redirección al cliente mediante un código 3XX.

Algunas situaciones en las que puede ser útil:

- El contenido de un servidor se ha movido a una URI (o URL) diferente.
- Convertir una solicitud POST en una solicitud GET.
- Redirigir la petición a otro proxy.
- Redirigir peticiones a Servlets, JSPs, ASPs, etc.

5.10.8. Cookies

Una cookie es un fragmento de información que envía un servidor web en una respuesta HTTP y es almacenada, si su configuración lo permite, por el navegador para su uso futuro. El navegador puede enviar la cookie en solicitudes posteriores al mismo servidor.


Los servidores envían las cookies usando cabeceras (Cookies y Set-Cookie) y pueden incluir los siguientes detalles:

- Nombre (Name). Nombre de la cookie.
- Contenido o valor (Value). Texto con el valor de la cookie.
- Fecha de expiración (Expires). Fecha/hora en la que la cookie deberá ser descartada por el navegador.
- Ruta (Path). Lugar donde la cookie será guardada por el navegador.
- Dominio (Domain). Nombre de dominio de donde "viene" la cookie.

5.10.8. Cookies

```
HTTP/1.1 200 OK
Date: Wed, 18 May 2011 11:23:39 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7
Set-Cookie: visitas=1; expires=Thu, 17-May-2012 11:23:39 GMT
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 87
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: text/html
```

Cookie



5.10.8. Cookies

Cuando un navegador realiza una solicitud a un servidor HTTP consulta las cookies que tiene almacenadas. Si existen cookies no caducadas cuya ruta y dominio coincidan con los de la petición, se las envía al servidor usando cabeceras (Cookies y Set-Cookie).

Las cookies son utilizadas por los servidores web para diferenciar usuarios y conexiones y actuar en consecuencia. Recuerda que HTTP es un protocolo "sin estado", cada transferencia de datos es independiente de la anterior sin ninguna relación entre ellas. ¿Cómo es posible que cuando accedes a una web, en la que introduces tu nombre de usuario y contraseña, el servidor web recuerde que eres tú mientras navegas por diferentes páginas, que implican varias peticiones HTTP (es lo que se conoce como sesión).

Actividades

- Accede a un navegador de tu equipo. Visita varias páginas web de Internet. Accede a las propiedades de configuración del navegador y busca cómo puedes consultar las cookies que tienes almacenadas. Comprueba qué cookies ha enviado las páginas que has visitado al iniciar la actividad.
- Las cookies en sí mismas no son un código malicioso, solo son texto, pero pueden ser utilizadas con fines malintencionados para obtener datos privados de usuarios como hábitos de navegación y sitios visitados, pueden ser robadas y/ o falsificadas para realizar ataques de seguridad, etc.
- Busca información en Internet sobre las nuevas normativas Europeas de cómo se controla y rastrea a los usuarios en Internet ¿Cómo se relaciona con las cookies?
- ¿En qué consiste la opción do-not-track que incorporan las nuevas versiones de los navegadores?

10.9. Autenticación

HTTP soporta el uso de mecanismos de autenticación para controlar el acceso a los recursos que ofrece el servidor. Estos mecanismos están basados en el uso del código de estado 401 y en las cabeceras WWW-Authenticate y Authorization.

Los navegadores muestran al usuario un cuadro de diálogo para que se identifique.

Algunos mecanismos son:

- Basic. El cliente envía un usuario y una clave codificados con el algoritmo base64. Método no seguro, es trivial capturar y obtener la clave enviada.
- Digest. El cliente envía un usuario y una función hash (resumen) de la clave al servidor (se utiliza el algoritmo MD5). Es más seguro que el método Basic pero es vulnerable a varios ataques y por lo tanto también es inseguro.



10.9. Autenticación

La autenticación que ofrece HTTP no es segura y por ello la autenticación se traslada a las aplicaciones web. En las aplicaciones web actuales la autenticación se basa en el uso de formularios XHTML en los que se introducen usuario y clave que son enviados al servidor (usando parámetros de los métodos GET o POST), donde son tratados por alguna aplicación escrita en un lenguaje como PHP, JSP, ASP, Ruby, etc., o en el uso de certificados digitales y HTTPS (Hypertext Transfer Protocol Secure)



10.10. Conexiones persistentes

El uso de conexiones persistentes en HTTP consiste en que varias peticiones y respuestas sean transferidas usando la misma conexión TCP. Su uso reduce el número de conexiones TCP, lo que repercute en un menor gasto de CPU /memoria y una reducción de los tiempos de respuesta.

11. MIME

MIME (Multipurpose Internet Mail Extensions) consiste en una serie de especificaciones orientadas a intercambiar en Internet, usando protocolos como HTTP y SMTP, todo tipo de recursos (texto, audio, vídeo, imágenes, ...) de forma transparente a los usuarios. Inicialmente, fueron usadas en el correo electrónico pero en la actualidad se ha generalizado su utilización a otros servicios.

Entre otros aspectos definen:


11. MIME

- Una serie de tipos y subtipos (por ejemplo text/html, image/gif, text/css, audio/x-mpeg, multipart/form-data, video/jpeg, ...) que determinan el contenido de los recursos enviados a través de la red. Son utilizados por los clientes y servidores para saber el tipo de recurso que reciben y cómo tienen que manejarlo.
- Un conjunto de reglas para codificar mensajes no ASCII.
- Un conjunto de cabeceras que son utilizadas por los protocolos para informar a los clientes y servidores sobre los recursos transmitidos:
 - MIME-Version. Versión de MIME usada.
 - Content-Description. Texto que describe el contenido.
 - Content-Id. Identificador único.
 - Content-Transfer-Encoding. Cómo se codifican los datos binarios usando ASCII.
 - Content-Type. Tipo y subtipo.

11. MIME

```
HTTP/1.1 200 OK
Date: Wed, 18 May 2011 11:23:39 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7
Set-Cookie: visitas=1; expires=Thu, 17-May-2012 11:23:39 GMT
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 87
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: text/html
```

Cabecera, tipo y subtipo MIME



5.12 Seguridad

HTTP no es un protocolo seguro:

- El intercambio de información se realiza en texto plano. Es vulnerable a ataques de análisis de tráfico de red (sniffing).
- Los mecanismos de autenticación como Basic y Digest no son seguros.
- No se usan mecanismos para garantizar que los equipos involucrados en la transferencia son quienes dicen ser. Es vulnerable a ataques de suplantación de identidad (spoofing y man-in-the-middle).
- Existen ataques que se basan en el robo o falsificación de cookies y/ o parámetros enviados en la URL o en el contenido de los mensajes, y que permiten al atacante "robar la identidad de un usuario" suplantarle en webs (bancos, webmails, redes sociales, ...).



Actividad

Busca en información en Internet sobre unos ataques web muy extendidos: SQL Injection.

Simula en tu servidor un ataque por SQL Injection.

13. Protocolo HTTPS

HTTPS (Hyper Text Transfer Protocol Secure) es un protocolo que utiliza SSL (Secure Sockets Layer) o en TLS (Transport Layer Security) para encapsular mensajes HTTP. Gracias a la utilización de algoritmos criptográficos y certificados digitales se puede garantizar, la confidencialidad y la integridad de la información transmitida, así como la autenticidad de los servidores.

- Los clientes utilizan https:// en las URIs (o URLs).
- Los servidores web por defecto escuchan peticiones HTTPS en el puerto 443/TCP.

En la actualidad todos los sitios web que manejen información personal y/o confidencial de los usuarios (correo electrónicos, redes sociales, bancos, comercio electrónico, ...) deberían usar HTTPS.

14. Alojamiento virtual de sitios web

El alojamiento virtual de sitios web (web virtual hosting) consiste en simular que existen varias máquinas (hosts) con sus respectivos sitios web sobre un solo servidor web, es decir, alojar varios sitios web (por ejemplo [www. asir. es](http://www.asir.es) y www.daw.es) en un mismo servidor.

También se usan los términos hosts virtuales, servidores virtuales, y sitios virtuales para referirse a este tipo de configuración.

El uso de servidores web virtuales permite reducir el número de máquinas físicas necesarias para alojar los millones de sitios web que existen en Internet y al mismo tiempo aprovechar mejor los recursos (uso de CPU, memoria, ...) de los equipos.

14.1. Alojamiento virtual basado en IPs

El servidor tendrá diferentes direcciones IP por cada servidor web virtual. Cada servidor virtual atenderá peticiones en una dirección IP diferente. A efectos de los usuarios es como si existiesen varios servidores web, uno en cada dirección IP. Para realizar esta configuración la máquina, donde se ejecuta el servidor web, debe tener varias tarjetas de red configuradas cada una con una dirección IP o hay que asignar varias direcciones IP (alias o interfaces virtuales) a una misma tarjeta de red (la mayoría de los sistemas operativos actuales soportan esta funcionalidad).

DIBUJO

14.2. Alojamiento virtual basado en nombres

El servidor permite alojar varios nombres de dominio (por ejemplo `www.asir.es` y `www.daw.es`) sobre la misma dirección IP. Cada servidor virtual atiende las peticiones de un nombre de dominio.

Hay que configurar un servidor DNS que asocie los nombres de dominio con la misma dirección IP.

Este tipo de alojamiento se definió a partir de la versión HTTP /1.1. El servidor web es capaz de diferenciar el nombre de dominio por el que pregunta el cliente porque en el mensaje de petición HTTP se envía la cabecera Host con el nombre de dominio.

Es la forma de alojamiento más utilizada. Al permitir alojar varios dominios en un único equipo y dirección IP, se ahorra tanto en número de equipos como en direcciones IP. Además simplifica y facilita la administración centralizada de los servidores.

14.2. Alojamiento virtual basado en nombres

Cabecera Host. Usada por el servidor web para saber qué servidor virtual (si está configurado) tiene que atender la petición

```
GET / HTTP/1.1
Host: www.todofp.es
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0.1) Gecko/20100101
Firefox/4.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Keep-Alive: 115
Connection: keep-alive
Cookie: __utma=41836455.1669009742.1292858328.1303907907.1305452850.5;
__utmz=41836455.1305452850.5.5.utmcsrc=iesgrancapitan.org|utmccn=(referral)|utmcmd=referral|utmcct=/blog04
```



14.3. Alojamiento virtual basado en puertos

Cada servidor virtual atiende peticiones en una dirección IP y/ o dominio:puerto diferentes.

Consiste en combinar el alojamiento basado en IP y/o en nombres con el uso de varios puertos a las escucha.