

SERVIDORES WEB HTTP)

Servicio web y la WWW (World Wide Web)

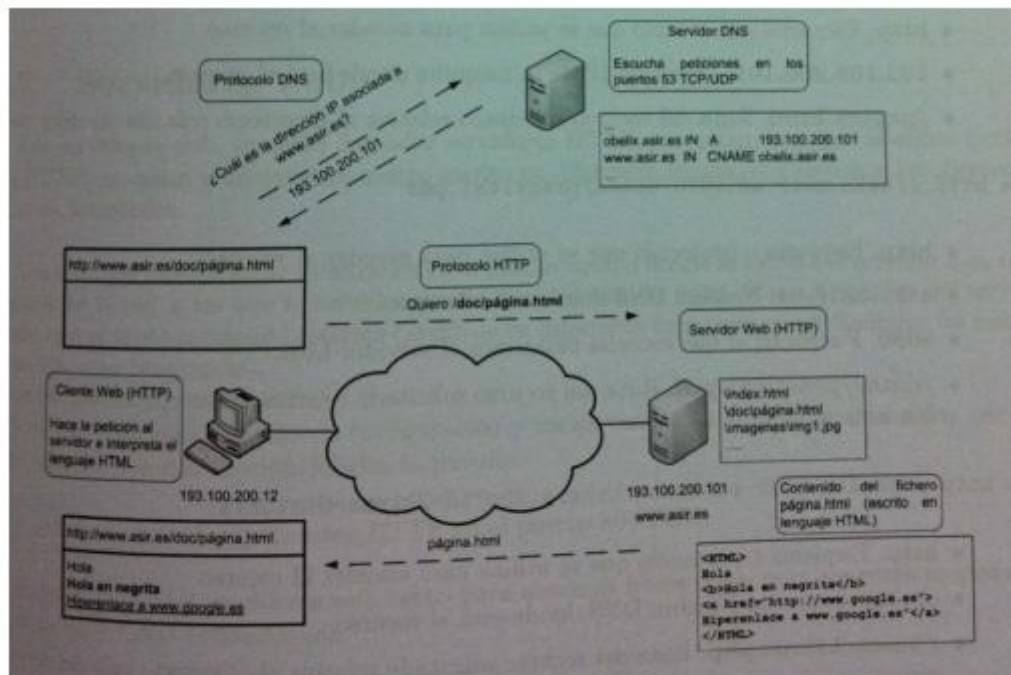
Se basa en el protocolo HTTP (Hyper Text Transfer Protocol) de capa de aplicación que facilita el **acceso a la información hipermedia remota** de sistemas conectado a la red TCP/IP.

La WWW es un servicio de distribución de información que permite acceder a recursos electrónicos y aplicaciones distribuidos en servidores por todo internet e identificados/localizados por direcciones (URIs o URLs).

Hiperenlaces

W3c controla actualmente el desarrollo de la WWW y desarrolla estándares WEB

Componentes y funcionamiento

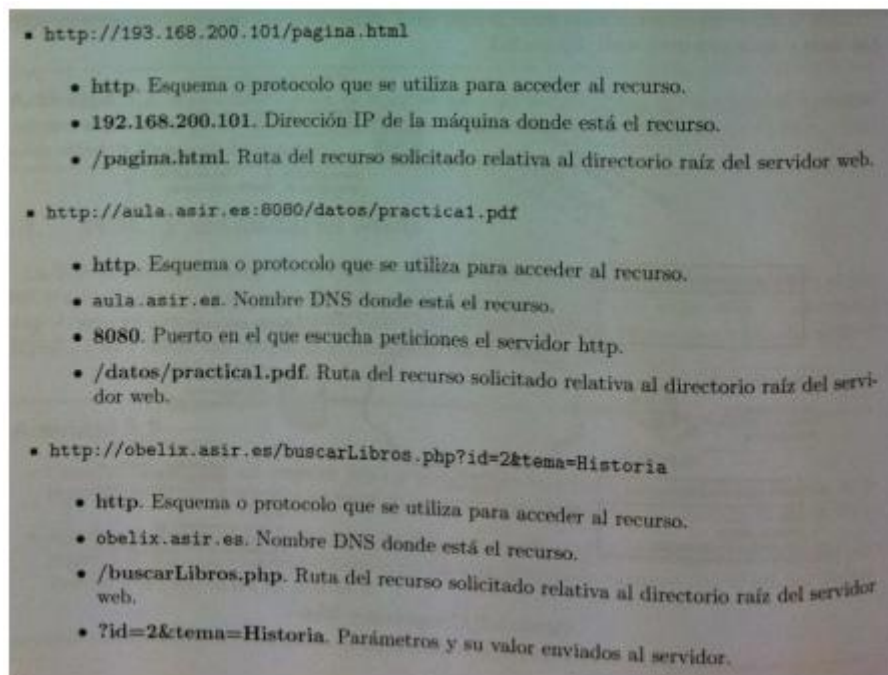


Nombres y direcciones (URIs y URLs)

Son cadenas de caracteres que se identifican de forma inequívoca los recursos que se acceden desde la web y que permiten su localización y acceso

URI, Uniform Resource Identifier -> Permite gran variedad de sistemas de denominación y métodos de acceso

URL, Universal Resource Locator -> Son tipos de URIs



Cientes web

También conocidos como navegadores, son programas con los que interactúa el usuario, que le permiten introducir URIs / URLs para acceder a recursos disponibles en la red.

Pueden actuar como clientes de diferentes protocolos, pero su uso principal es como cliente HTTP.

Reciben los recursos de los servidores web, los procesan y muestran al usuario, permitiéndole la interacción si es necesario

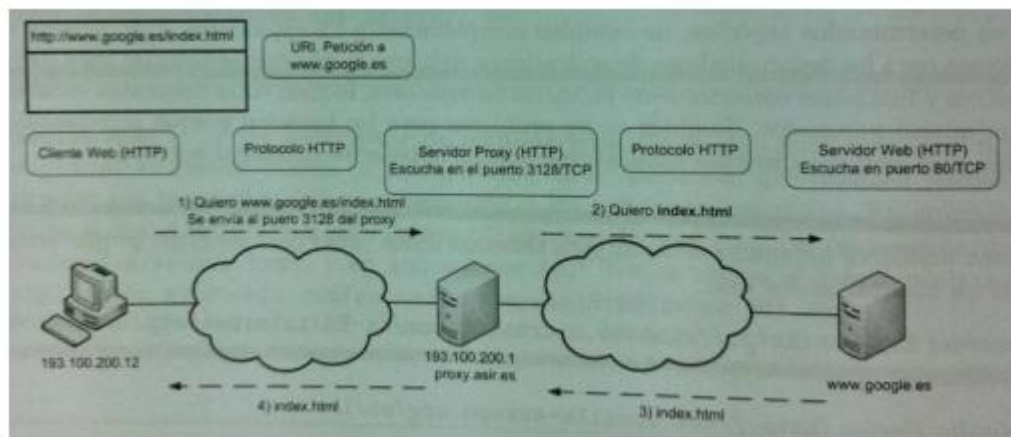
Mantienen una memoria cache para la información de uso reciente

Proxies web / proxy HTTP

Hace de intermediario entre un cliente web y un servidor web

Proxy directo (forward proxy): recibe una petición iniciada por un cliente web y se la traslada al servidor web. La petición es hacia el servidor web, no hacia el proxy que solo hace de intermediario.

Su principal cometido es asegurar y controlar el acceso a redes externas de los clientes (web) de una organización



Proxy inverso (reverse proxy): recibe una petición de un cliente web y se la reenvía a uno o varios servidores web. La petición es hacia el proxy (que para los clientes es un servidor normal).

Se usan para dar acceso a servidores web que están detrás de un firewall y no son accesibles directamente, para balancear la carga entre varios servidores web, para optimización de tiempos de respuesta mediante el uso de cache

Protocolo HTTP (HyperText Transfer Protocol)

Es de facto el protocolo de comunicación en la web

Define las reglas de comunicación entre los componentes software involucrados (clientes, servidores y proxies)

Es un protocolo sin estado que utiliza TCP como protocolo de transporte (no en http/3 que usa UDP)

Determina los tipos de peticiones que los clientes pueden enviar, así como el formato y la estructura de las respuestas

También define una estructura de metadatos que se incluyen en las cabeceras y que se envían tanto en las peticiones como en las respuestas

Funcionamiento básico HTTP

El usuario introduce una URI (o URL) en el navegador o hace clic en un hipervínculo

El navegador establece una conexión TCP con el servidor web (si se invoca con nombre de dominio previamente debe resolverla). Si no se indica número de puerto se conecta al puerto 80 del servidor

Una vez establecida la conexión TCP, el navegador envía un mensaje HTTP de petición

El servidor envía un mensaje de respuesta que depende de la petición realizada y del estado del servidor

Se cierra la conexión TCP

Códigos de estado y error HTTP:

Códigos que envía el servidor en su respuesta HTTP al cliente y que informan de como ha sido procesada la petición

100-199: informativos

200-299: éxito

300-399: redirección

400-499: errores del cliente

500-599: errores en el servidor

Almacenamiento en cache (caching)

Los servidores web contienen información que no cambia durante ciertos periodos de tiempo, algunos de ellos en periodos muy largos

HTTP soporta almacenamiento en cache para evitar tráfico innecesario y aumentar el rendimiento

Así los navegadores y proxies pueden almacenar localmente recursos durante un tiempo y evitar nuevas descargas innecesarias

Definen cabeceras que permite controlar lo que se puede almacenar en cache y si se puede cachear en un proxy

Compresión

Es posible que los servidores compriman los recursos antes de enviarlos a los clientes (usando gzip por ejemplo) para reducir el tráfico de red

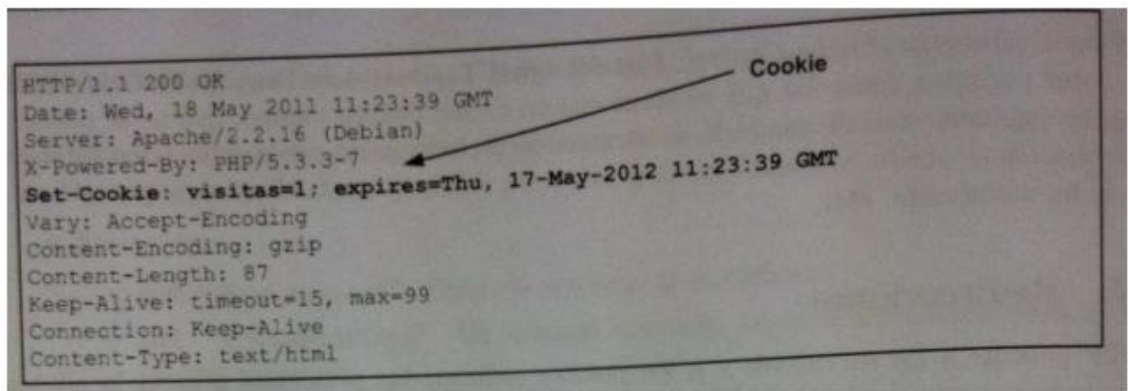
Los clientes usan cabeceras en los mensajes de petición para indicar que soportan compresión

Los servidores en las respuestas también usan cabeceras para indicar al cliente que envían datos

Cookies

Una cookie es un fragmento de información que envía un servidor web en una respuesta HTTP y es almacenada por el navegador para su uso futuro en solicitudes posteriores al mismo servidor

Los servidores envían las cookies usando cabeceras



Cuando un navegador utilizar una nueva solicitud a un servidor HTTP consulta las cookies que tiene almacenadas y se las envía al servidor usando cabeceras

Las cookies son utilizadas por los servidores para segmentar usuarios y conexiones

Permite, por ejemplo, que, aunque HTTP sea un protocolo sin estado recuerde datos de la navegación durante todo el tiempo que dura la misma

Permite también que se muestren anuncios, productos, noticias,... relacionadas con las preferencias del usuario

Autenticación

HTTP soporta mecanismos de autenticación para el control del acceso a los recursos que ofrece el servidor

No obstante, la autenticación que ofrece no es segura y por ello se traslada a las aplicaciones web

Conexiones persistentes

El uso de conexiones persistentes en HTTP consiste entre varias peticiones y respuestas sean transferidas usando la misma conexión TCP, lo que reduce el número de conexiones y un menor consumo de recursos de CPU y memoria y consiguiente reducción de tiempos de respuesta

En HTTP 1.1 se usan conexiones persistentes por defecto (no así en HTTP 1.0)

MIME (Multipurpose Internet Mail Extensions)

Consiste en una serie de extensiones orientadas al intercambio de todo tipo de recursos (texto, audio, video, imágenes, ...) en internet usando protocolos HTTP y SMTP. Inicialmente fueron usadas para el correo electrónico, pero en la actualidad se extendió su uso a otros servicios.

Definen:

Tipos y subtipos que determinan el contenido de los recursos enviados:
text/HTML, text/CSS, image/gif, audio/x-mpeg, video/mpeg-2, multipart/from-data

Un conjunto de reglas para codificar mensajes no ASCII

Un conjunto de cabeceras para informar a los clientes y servidores sobre los recursos transmitidos

Se utilizan las cabeceras de tipo MIME en:

Los mensajes de respuesta para informar al cliente de los recursos que le llegan. El navegador en función del tipo MIME visualiza el recurso, lo abre con una aplicación externa o pregunta al usuario como proceder

Los mensajes de petición de los clientes para informar al servidor de los MIME que aceptan

Para encapsular uno o mas recursos en el cuerpo del mensaje utilizando un tipo MIME denominado multipart