

2. Protocolo de transferencia de hipertexto (HTTP)

HTTP (HyperText Transfer Protocol) tiene como objetivo transferir contenido entre dos nodos.

Procesos involucrados:

- Cliente. Proporciona una interfaz de usuario, generalmente un navegador, que permite al usuario obtener y representar correctamente el contenido.
- Servidor. Extrae y proporciona al cliente el contenido procedente de bases de datos, sistemas de archivo, dispositivos u otras fuentes.

HTTP funciona sobre TCP, puerto 80 y, alternativamente, puerto 8080.

El contenido se puede codificar en una gran variedad de formatos, algunos de los cuales integran texto, audio, vídeo y otros. Sin embargo, el protocolo HTTP es independiente de la codificación del contenido (PDF, HTML, XML, ASCII, JPEG, MP3, etc.). La representación adecuada del contenido en el lado del cliente depende de las capacidades del software del cliente.

Lenguaje de marcas hipertexto (HTML, Hypertext Markup Language) y la Web

HTML y sus estándares relacionados (CSS, Javascript, etc.) permiten diseñar contenido mediante la especificación de la estructura y apariencia de la información.

Todo el contenido en Internet relacionado a través de hipervínculos constituye una gran red de contenidos que se conoce comúnmente con el nombre de WWW.

Localizador uniforme de recursos (URL, Uniform Resource Locator)

Una URL o dirección web es un texto que indica como acceder a una página web y obtener su contenido. La URL puede indicar la siguiente información: nombre del esquema (protocolo), nombre del host (o dirección IP), número de puerto, nombre de usuario, contraseña, ruta, consulta (parámetros) y fragmento (parte o sección del contenido).

esquema: [/ [usuario:contraseña@] host[:puerto]] [/ruta[?consulta][#fragmento]

Las URL se utilizan en hipervínculos y también como datos de entrada para navegadores y otros clientes HTTP.

Extensiones multipropósito de correo de Internet (MIME, Multipurpose Internet Mail Extensions)

Los navegadores también pueden iniciar otras aplicaciones disponibles para mostrar otros tipos de contenido. Para que el navegador pueda realizar el tipo de procesamiento que corresponda, el tipo de contenido debe especificarse en la transferencia HTTP. Para este propósito, se utiliza parte del llamado estándar MIME.

El estándar MIME permite indicar un tipo y subtipo estándar (tipos MIME o Media Types). Por ejemplo:

- texto/html, contenido en formato HTML;
- video/mpeg, video producido en formato MPEG;
- image/*, imagen en cualquier formato (GIF, JPEG, PNG, BMP, etc.).

La lista completa es mantenida y controlada por la IANA (Internet Assigned Numbers Authority), que ha definido también un procedimiento de registro de nuevos tipos. Esta lista se puede encontrar en <https://www.iana.org/assignments/media-types/media-types.xhtml>.

2.1. Operación y formato de mensaje

Operación

En una conexión HTTP 1.0, solo se intercambian dos mensajes: el cliente inicia la conexión y envía una solicitud; a continuación el servidor devuelve una respuesta (la respuesta puede incluir el recurso solicitado, un mensaje de error u otra información); finalmente, cliente y servidor cierran la conexión.

Formato del mensaje

Cliente y servidor se comunican mediante mensajes de texto sin formato (ASCII). El cliente envía solicitudes (requests) al servidor y el servidor envía las respuestas (responses) correspondientes.

Mensaje de solicitud

El mensaje de solicitud consta de:

- una línea de solicitud (por ejemplo, GET /images/logo.png HTTP/1.1),
- cabeceras de solicitud (por ejemplo, Accept-Language: en),
- una línea en blanco y
- un cuerpo de mensaje opcional.

Mensaje de respuesta

El mensaje de respuesta consta de:

- una línea de estado que incluye el código de estado y descripción (por ejemplo, HTTP/1.1 200 OK),
- cabeceras de respuesta (por ejemplo, Content-Type: texto/html),
- una línea en blanco y
- un cuerpo de mensaje opcional.

Métodos de solicitud

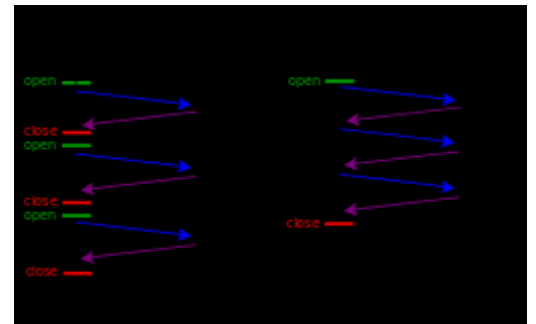
HTTP define varias operaciones o métodos para indicar el procesamiento que se realizará sobre el recurso especificado. Los métodos más comunes son los siguientes:

- GET. El método GET solicita la obtención del recurso especificado; las solicitudes GET solo deberían recuperar datos y no deberían tener otro efecto.
- POST. El método POST permite el envío de datos para que sean incorporados al recurso especificado; los datos publicados (POSTed) pueden proceder, por ejemplo, de un formulario HTML.
- HEAD. El método HEAD solicita una respuesta idéntica a la de una solicitud GET, pero omitiendo el cuerpo de la respuesta; esto es útil para obtener metadatos disponibles en encabezados de respuesta evitando transferir todo el contenido.

Mejoras de operación en HTTP 1.1

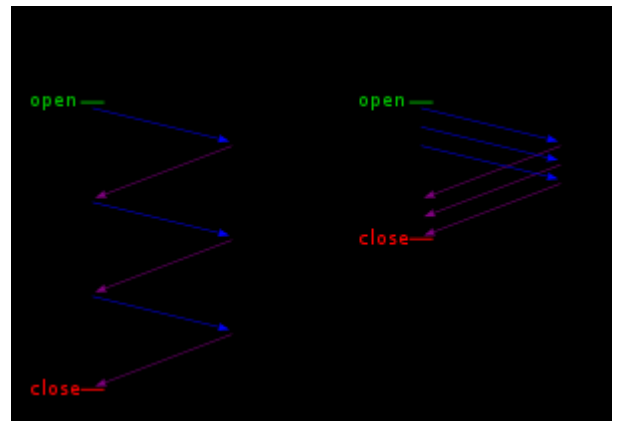
Keep-alive

Permite utilizar una conexión para más de una solicitud y respuesta. Para ello, las conexiones se mantienen abiertas durante un tiempo después de transferir la respuesta. El mecanismo reduce perceptiblemente la latencia de la comunicación, porque el cliente no necesita renegociar la conexión TCP (handshake) una vez enviada la primera solicitud.



Pipelining

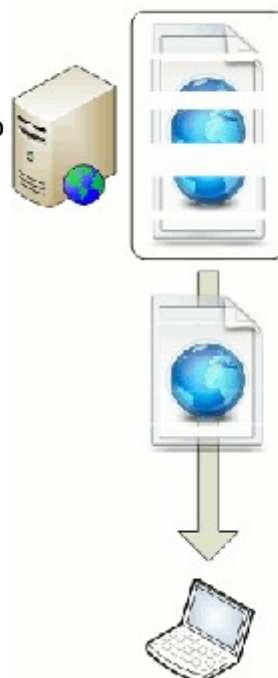
Permite a los clientes enviar múltiples solicitudes sin esperar la recepción de cada respuesta. De esta forma, el tiempo de carga puede mejorar especialmente en medios de transmisión de alta latencia, como las transmisiones vía satélite.



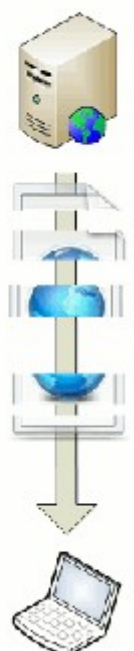
Chunked transfer encoding

Se puede usar en conexiones keep-alive cuando el servidor desconoce la longitud total del contenido a enviar, y por tanto, no lo puede indicar al comienzo de la respuesta (el cliente requiere esta información para saber cuándo ha finalizado la respuesta). En este modo, el servidor divide el flujo de datos en fragmentos independientes de tamaño conocido, y cada uno se envía precedido por su longitud correspondiente.

Store-and-Forward



Chunked Encoding



Byte-serving

El servidor transmite solo la parte de un recurso solicitada explícitamente por el cliente. Por ejemplo, el cliente puede hacer uso de este modo en los casos en que un archivo grande se haya entregado solo parcialmente y se necesite una porción del archivo. Se trata, por lo tanto, de un método de optimización del ancho de banda.

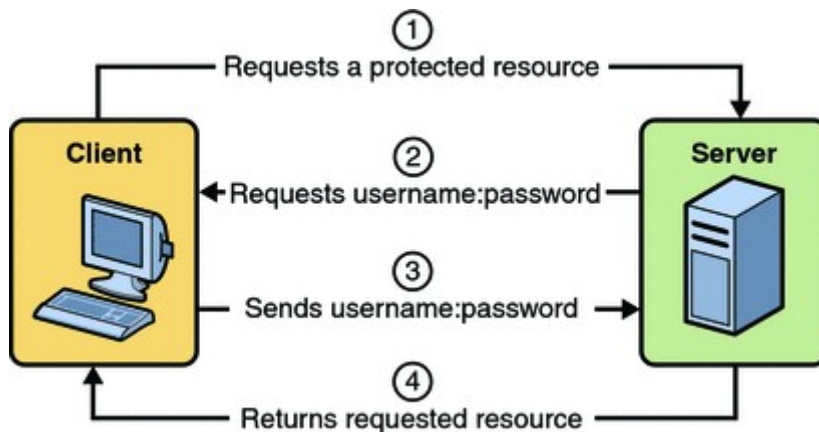
Cookies

Una cookie HTTP es un dato identificativo establecido por el servidor en la respuesta y almacenado por el cliente en el nodo. El cliente envía estos datos al servidor correspondiente en solicitudes posteriores, por lo que el servidor puede agrupar una colección de solicitudes independientes como pertenecientes al mismo usuario.

Las cookies constituyen la base de múltiples aplicaciones actuales: productos añadidos al carrito de la compra en tiendas online, seguimiento de la actividad de navegación del usuario o establecimiento de sesiones, entre muchas otras.

Autenticación

HTTP proporciona varios esquemas de autenticación. El enfoque más simple es la llamada autenticación básica.



3. Preprocesador de hipertexto (PHP)

Esta unidad es una introducción al lenguaje de programación PHP a través de una serie de ejercicios basados en la información disponible en el [sitio oficial](#). Para llevarlos a cabo es necesario revisar las referencias que se proponen en cada sección de la unidad. Generalmente, cada ejercicio propone la realización de un programa sencillo, y opcionalmente algunas cuestiones. En cada uno debe figurar el código fuente que resuelve el problema propuesto, debidamente comentado cuando revista alguna dificultad, así como la respuesta a las cuestiones planteadas. Además, es necesario adjuntar el archivo o archivos *.php para que la solución proporcionada se pueda probar.