

HTTP y HTTPS

Sistemas informáticos en red

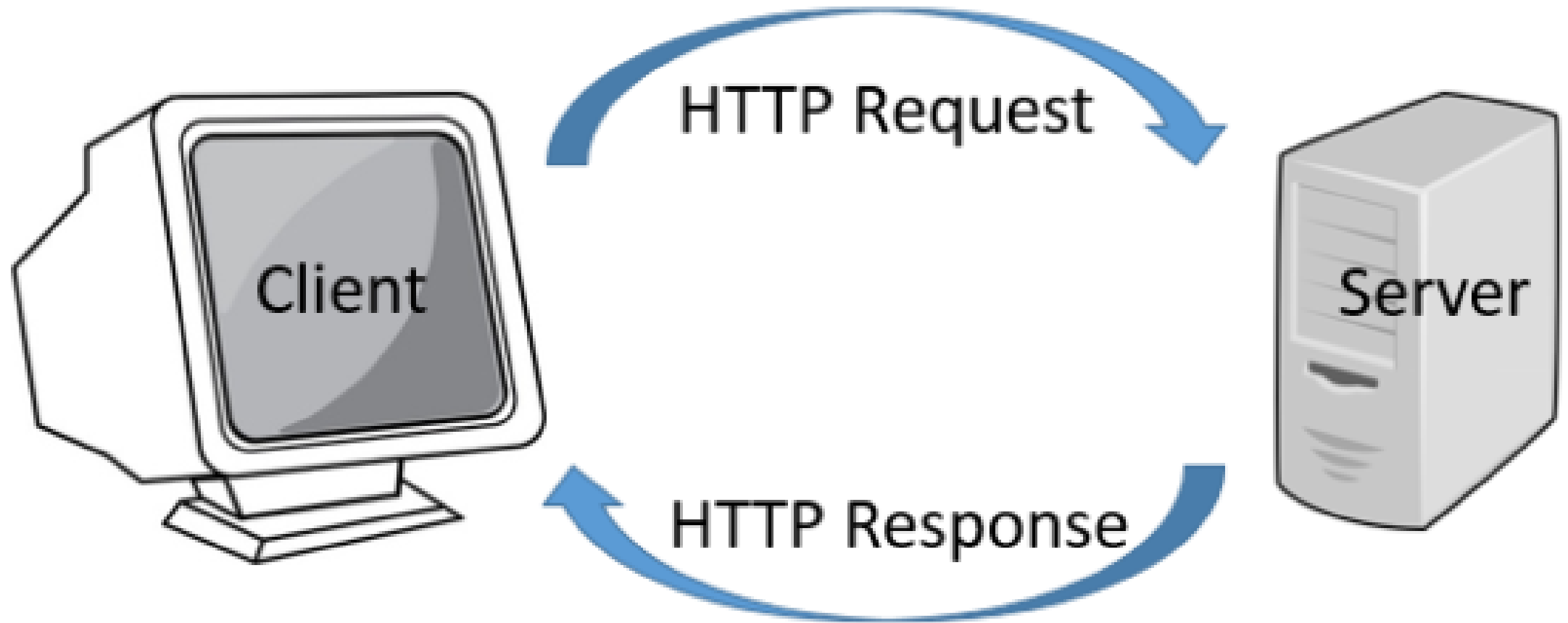
Introducción

- **Hypertext Transfer Protocol (HTTP)** es un protocolo de la capa de aplicación.
- Fue diseñado para la **comunicación de hipertexto** (HTML) a través de la **World Wide Web**.
- En la actualidad se utiliza como protocolo de comunicaciones en red para **todo tipo de formatos** de texto (XML, Json, texto plano...) e incluso otros formatos binarios.
- Es posible habilitar la encriptación de las comunicaciones utilizando el protocolo **SSL**, lo que da lugar al protocolo **HTTPS**
- Fue documentado por primera vez en 1997.
- En la actualidad se utiliza la versión 2 aunque ya está en marcha la propuesta de la versión 3.

Funcionamiento de HTTP

- Funciona como un protocolo de petición-respuesta (**request-response**), es decir, existe un emisor que envía una petición al receptor, el cual responde con una respuesta.
- Por ejemplo, un navegador solicita a un servidor que se le proporcione una determinada dirección Web, el servidor responde enviando un documento HTML que es mostrado en el navegador.
- La comunicación HTTP está basada en el protocolo **TCP** por defecto.
- El puerto por defecto para las comunicaciones **HTTP** es el **80**.
- El puerto por defecto para las comunicaciones **HTTPS** es el **443**.

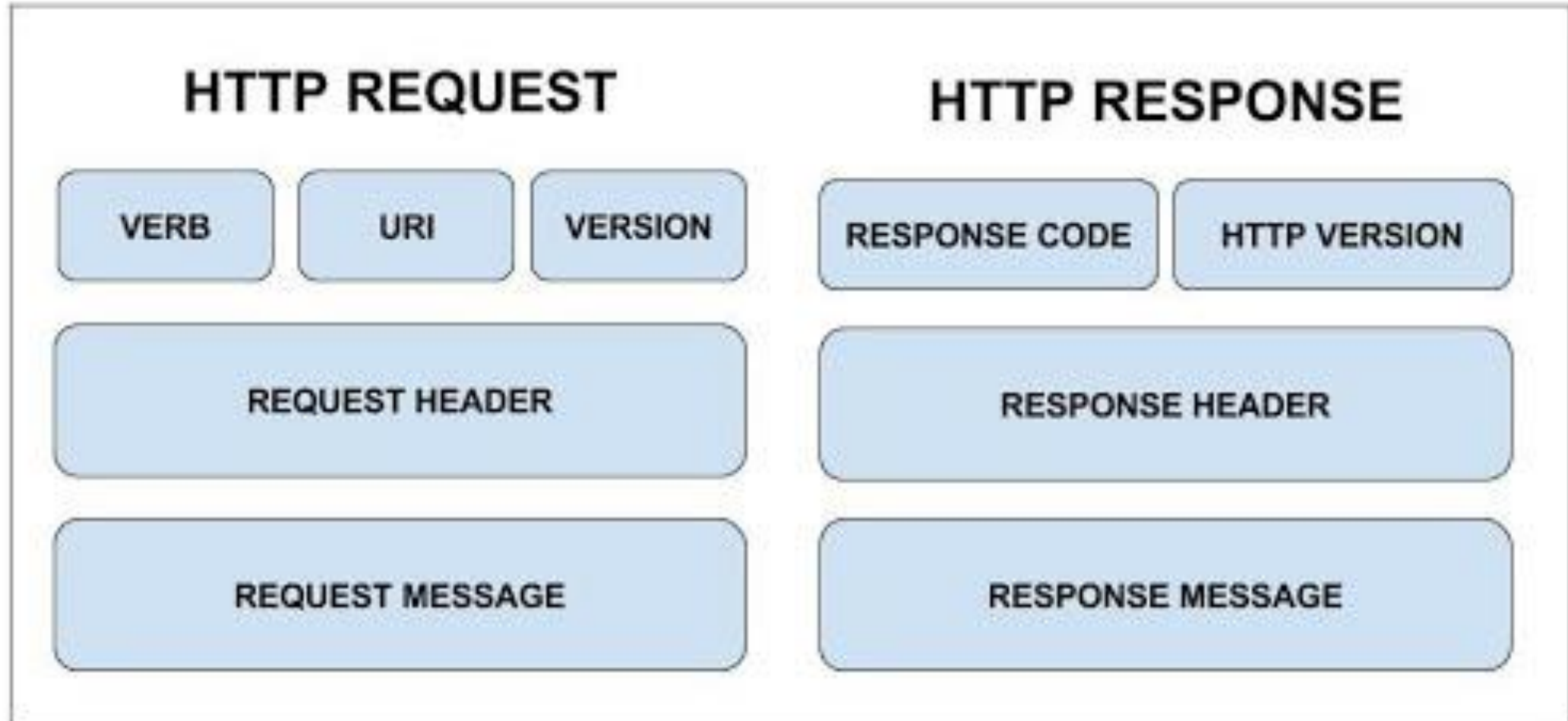
Funcionamiento de HTTP



Funcionamiento de HTTP

- Es un protocolo sin estado (**stateless**), es decir, el servidor no mantiene información del usuario ni su estado durante la duración de múltiples peticiones del mismo.
- Por esto, es necesario la utilización de **cookies** para mantener la información de sesión de los usuarios.
- Provee de esquemas de autenticación de usuarios que facilitan esta tarea.
- Cliente y servidor intercambian mensajes durante su comunicación:
 - **Request message**
 - **Response message**

Mensajes HTTP



HTTP Methods

- Existen diferentes acciones que un cliente puede desear hacer cuando se comunica con un servidor: crear un recurso, modificar o eliminar uno existente, solicitar datos de un recurso, solicitar listados de recursos, etc...
- Para poder diferenciar mejor qué **tipo de acción** desea realizar el cliente, existen un conjunto de métodos HTTP disponibles que se pueden usar en una petición.
- Cada **petición** HTTP utiliza **un solo método** HTTP.
- Se denomina **semántica del método** al significado que se le da a cada acción, es responsabilidad del programador utilizar coherentemente los distintos métodos dada su semántica.

HTTP GET

- Solicita una representación del recurso especificado.
- Semánticamente, las solicitudes que usan GET solo deben recuperar datos y no deben tener ningún otro efecto.

The diagram illustrates the structure of an HTTP GET request. It shows a request line and a block of request headers. Brackets above the request line identify its parts: 'The HTTP Method' for 'GET', 'Path to the source on Web Server' for '/RegisterDao.jsp', 'Parameters to the server' for '?user=ravi&pass=java', and 'Protocol Version Browser supports' for 'HTTP/1.1'. A bracket to the left of the headers identifies them as 'The Request Headers'.

The HTTP Method

Path to the source on Web Server

Parameters to the server

Protocol Version Browser supports

GET /RegisterDao.jsp?user=ravi&pass=java HTTP/1.1

The Request Headers

Host: www.javatpoint.com
User-Agent: Mozilla/5.0
Accept-text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Keep-Alive: 300
Connection: keep-alive

▼ General

Request URL: https://www.google.es/

Request Method: GET

Status Code: 🟢 200

Remote Address: 142.250.201.67:443

Referrer Policy: strict-origin-when-cross-origin

▶ Response Headers (13)

▼ Request Headers

:authority: www.google.es

:method: GET

:path: /

:scheme: https

accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

accept-encoding: gzip, deflate, br

accept-language: en-US,en;q=0.9,es-ES;q=0.8,es;q=0.7

cookie: __Secure-3PSID=3QejAGvtWtfe-X2r-JKamfnWP2hVNM6sY5bLXFe1nHtLLijaKUavtzNwVrY6yESyyC7VNQ.;
__Secure-3PAPISID=FGTHZ3MF-Shu8F0p/AJvm3X2Ti40jd5SsH; NID=204=v0zAy10fvM197TPnJp_JSByK5X7SQMOUA
Cmprbnsf7VBNAleJqZbbUMR5JtG9lULR5LaLPzDlJanKI3s50N6oclta4trfB0higRTqSBLiTT0p4Q8R5bZWoeaMxJpID0
JTMtYTTHXvls8BUNcCxFedVIh6H2QLdaJeduiXj953w

sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"

sec-ch-ua-mobile: ?0

sec-fetch-dest: document

sec-fetch-mode: navigate

sec-fetch-site: none

HTTP HEAD

- Solicita metadatos del recurso especificado.
- Semánticamente, devuelve una respuesta idéntica a GET pero sin el cuerpo de la respuesta.
- Únicamente devuelve las cabeceras de la respuesta.

HTTP POST

- Envía datos para que sean procesados por el recurso identificado en la URI de la línea petición.
- Los datos se incluyen en el cuerpo de la petición
- Semánticamente, está orientado a crear nuevos recursos.
- Los datos pueden viajar en múltiples formatos, algunos de los más típicos son:
 - JSON
 - XML
 - Texto plano
 - Campos codificados en URL

HTTP PUT

- Envía datos para que sean actualizados en el recurso identificado en la URI de la línea petición.
- Los datos se incluyen en el cuerpo de la petición
- Semánticamente, está orientado a la modificación de un recurso existente.
- Los datos pueden viajar en múltiples formatos, algunos de los más típicos son:
 - JSON
 - XML
 - Texto plano
 - Codificados en la URL

Otros métodos HTTP

- **DELETE:** Sirve para eliminar el recurso especificado en la URL.
- **OPTIONS:** Devuelve los métodos HTTP que el servidor soporta para la URL especificada.
- **PATCH:** es similar al PUT, se utiliza PATCH cuando se quiere modificar parcialmente el recurso, es decir solo algún campo.
- **CONNECT:** Se utiliza para comprobar si hay conexión con el host

Mensajes de respuesta HTTP

- Todo mensaje de respuesta consta de lo siguiente:
 - Status code: código que representa el estado de la respuesta
 - Cabeceras de respuesta: array con cabeceras
 - Una línea en blanco
 - Opcionalmente, un cuerpo de la respuesta con datos (message body)
- Por ejemplo, cuando realiza una petición GET para descargar una página web recibiremos una respuesta con código 200, una serie de cabeceras y el documento HTML en el cuerpo.

Códigos de respuesta

- El código de respuesta es un número entero que tiene un significado establecido.
- **1xx**: Respuestas informativas. Indica que la petición ha sido recibida y se está procesando.
- **2xx**: Respuestas correctas. Indica que la petición ha sido procesada correctamente.
- **3xx**: Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
- **4xx**: Errores causados por el cliente. Indica que ha habido un error en el procesamiento de la petición a causa de que el cliente ha hecho algo mal.
- **5xx**: Errores causados por el servidor. Indica que ha habido un error en el procesamiento de la petición a causa de un fallo en el servidor.