



Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>

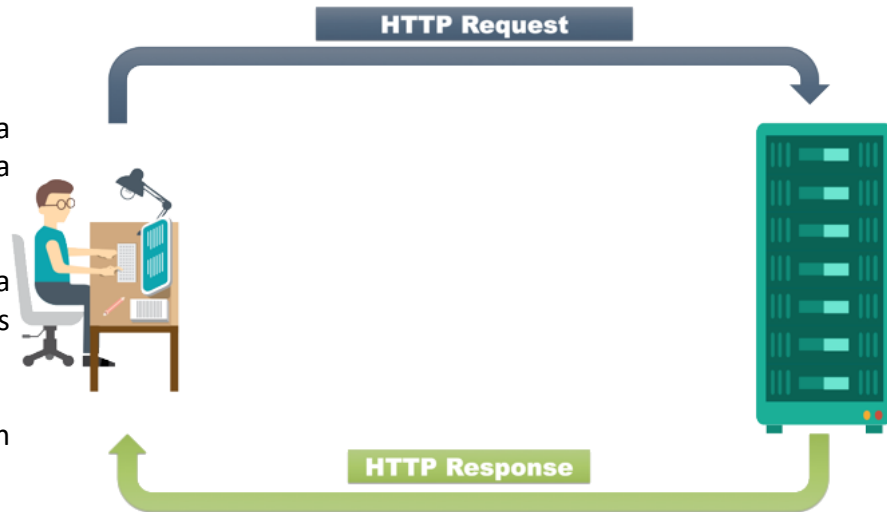


Protocolo HTTP

Es un protocolo cliente-servidor, lo que significa que el cliente envía una petición (request) al servidor y espera un mensaje de respuesta (response) del servidor.

Es un protocolo **sin estado**, lo que significa que el servidor no guarda información del cliente, cada petición es independiente de las demás.

Las peticiones se realizan a través de verbos que definen la acción que se va a realizar en el recurso del servidor.

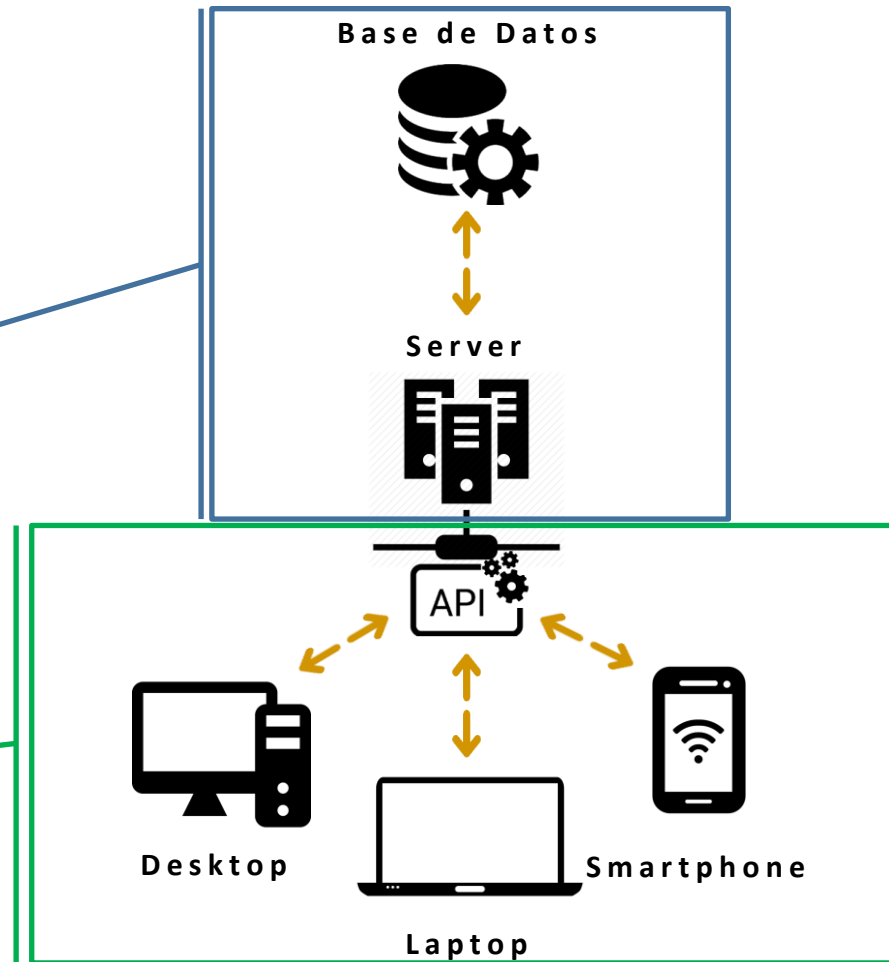


La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Cliente - Servidor

Una aplicación servidor es el elemento de la comunicación que responde a las peticiones de los clientes, proporcionando el servicio requerido

Una aplicación cliente es el elemento de la comunicación que pide o solicita un servicio de red

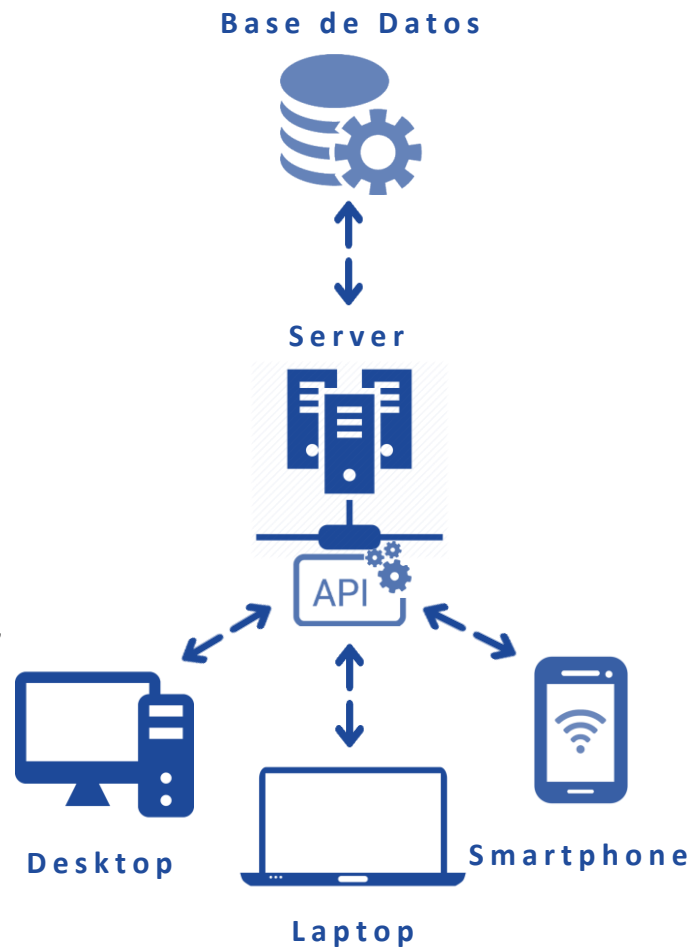


Ventajas

Escalabilidad : Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden agregar nuevos nodos a la red (clientes y/o servidores).

Fácil mantenimiento : Al estar distribuidas las funciones y responsabilidades entre varias computadoras independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por el cambio. Hoy día es frecuente también tener servidores en la nube

Seguridad: Permite administrar permisos de forma mas localizada.



Request

Para poder hacer las peticiones el protocolo HTTP proporciona una serie de métodos llamados verbos para realizar una acción en el recurso determinado.

Los verbos mas utilizados son:

GET: solicita una representación de un recurso específico. Las peticiones que usan el verbo GET sólo deben recuperar datos ya que la información que se envía viaja a través de la URL.

HEAD: pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

POST: se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT: reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE: borra un recurso en específico.



Response

Representa un recurso que generalmente es un HTML pero puede ser de cualquier otro tipo.

Además de devolver el recurso solicitado, si todo salió de forma exitosa, la respuesta devuelve siempre un código que nos provee mas información.

- Respuestas informativas (100–199)
- Respuestas satisfactorias (200–299)
- Redirecciones (300–399)
- Errores de los clientes (400–499)
- Errores de los servidores (500–599)

Los mas comunes que normalmente vemos:

404: Recurso No encontrado.

500: Error en el Servidor



URL (Uniform Resource Locator)

Localizador de Recursos Uniforme no es más que una dirección que es dada a un recurso único en la Web. En teoría, cada URL valida apunta a un único recurso.

Esquema El esquema define el protocolo a utilizar por ejemplo http:, https:, mailto:, ftp:, etc.

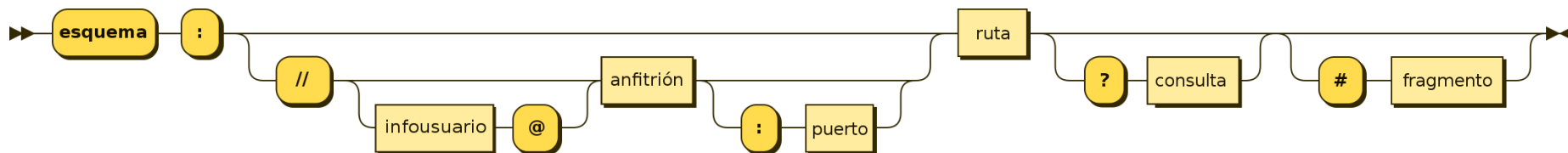
Host La IP o el nombre del servidor (Dominio) que se quiere acceder.

Puerto El puerto en el que está escuchando el servidor HTTP. Si se omite se asume que es el 80.

Ruta La ruta al recurso que se quiere acceder.

Consulta Contiene información adicional para el servidor en forma de propiedades (atributo=valor). Las propiedades se separan por un ampersand "&".

Fragmento La referencia a una ubicación interna del documento.



Client-Side Processing

En el desarrollo web, el "lado del cliente" también conocido como **FrontEnd** se refiere a todo en una aplicación que se muestra o tiene lugar en el cliente (dispositivo del usuario final). Esto incluye lo que ve el usuario, como texto, imágenes y el resto de la interfaz de usuario, junto con las acciones que realiza una aplicación dentro del navegador del usuario.

Los mas utilizado para el desarrollo del lado del cliente es HTML para darle estructura, CSS para estilizar y JavaScript para la lógica y validación.

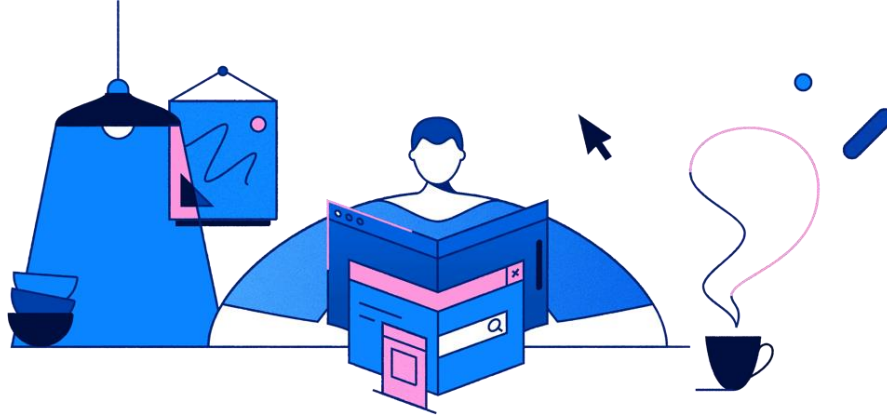


Navegador



Un navegador web es un software que te lleva a cualquier lugar de internet, de este modo podemos decir que un navegador representa en la mayoría de los casos al cliente, el cual solicita pedidos al servidor. Adicionalmente, tiene la responsabilidad de “comprender” las respuestas provenientes del servidor, interpretarlas y visualizarlas en pantalla de modo gráfico.

Es importante entender que los navegadores solo saben procesar texto plano, HTML, CSS y JavaScript.



HTML

Hyper Text Markup Language (lenguaje de marcas de hipertexto) es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página



CSS

Las hojas de estilo en cascada (CSS, cascading style sheets) es un lenguaje de reglas de estilo que usamos para aplicar forma a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.



JavaScript

Es un lenguaje de programación interpretado que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.



Server-Side Processing

En el desarrollo web, el "lado del servidor" también conocido como **BackEnd** significa todo lo que sucede en el servidor, en lugar de en el cliente. Esto incluye los datos y lógicas que mostrara el cliente al recibir las respuestas por nuestros servidores.

Las tecnologías utilizadas en el **lado del servidor** son muchos mas diversas que con **las del lado del cliente**.

