



Ing. en Desarrollo y Gestión de Software

ARQUITECTURAS DE SOFTWARE

Hugo Eden Maroyoqui Alvarado.

Team:

Rodríguez Díaz Carlos.

Coronel Ramírez Jhoana Abigail.

Sánchez Rendón Jesus Manuel.



INTRODUCCIÓN

La arquitectura **cliente-servidor** es un modelo de red en el que cada proceso se ejecuta una petición, servidor o un cliente.

Los **servidores** son unas maquinas robustas que se dedican a administrar información, las unidades de disco y el tráfico de la red.

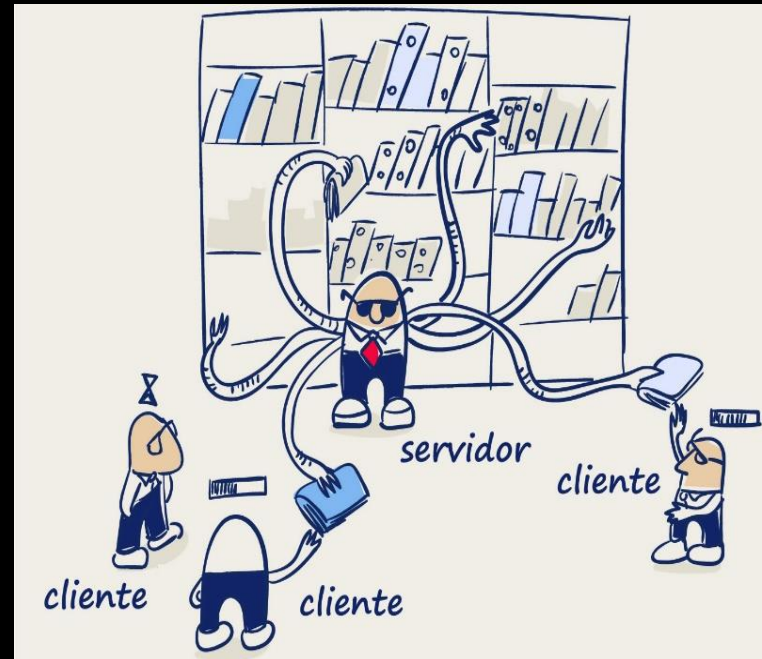
Los **clientes** dependen principalmente de los servidores para obtener recursos.





¿Qué es la arquitectura cliente-servidor?

Una arquitectura o modelo cliente-servidor es una **red de aplicaciones** que separa las tareas entre los clientes y los servidores que están establecidos en el mismo sistema o tienen que comunicarse a través de una red informática.

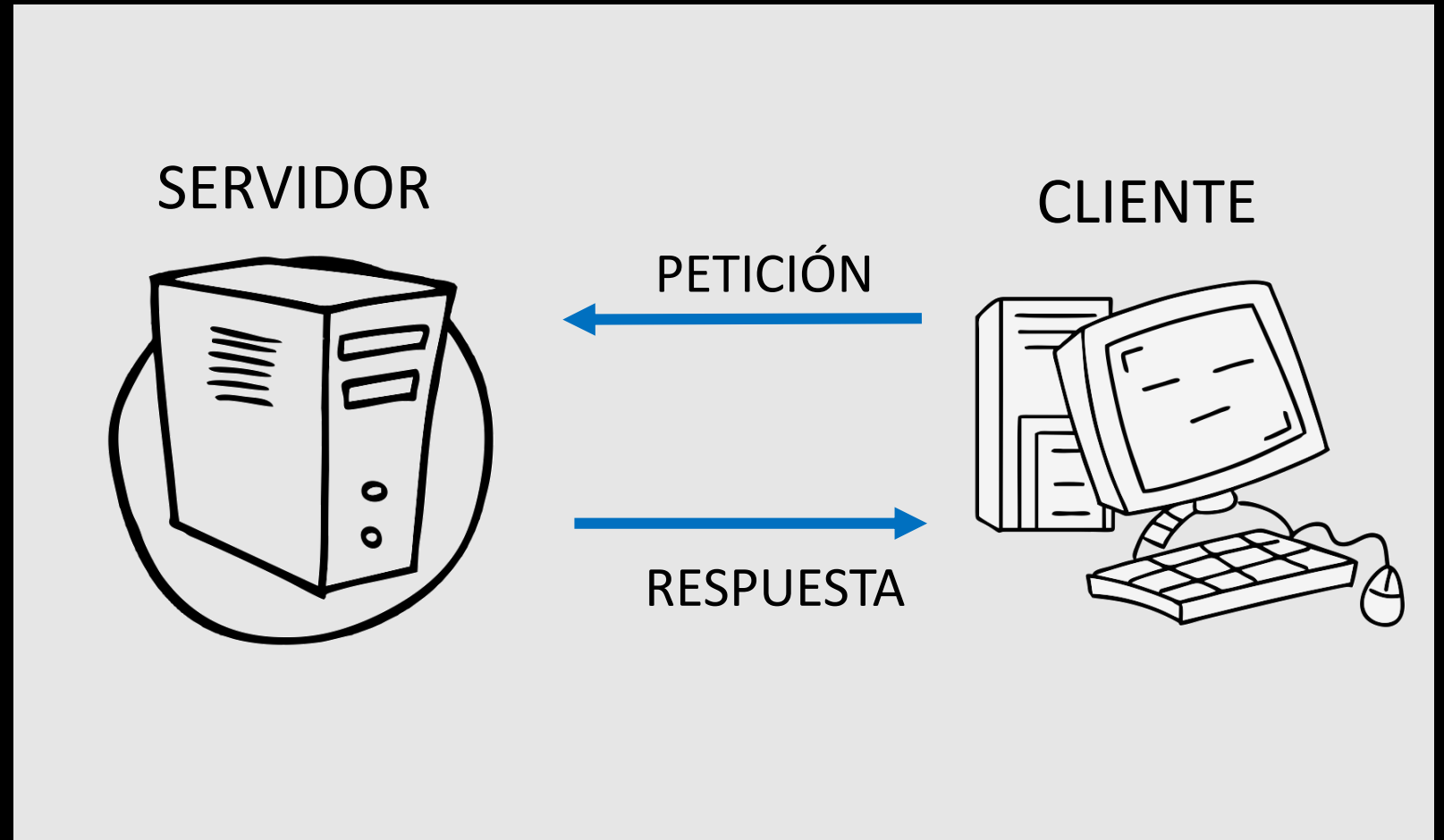


El **servidor-cliente** se basa principalmente en enviar la **solicitud a otro programa** para acceder al servicio que el servidor hace accesible.

Este servidor ejecuta algunos programas que comparten los recursos y distribuyen el trabajo entre los clientes.



¿Cómo funciona la arquitectura cliente-servidor?



Tipos de arquitecturas cliente servidor



Arquitectura de dos capas

Esta se utiliza para describir los sistemas cliente servidor en donde el cliente solicita recursos y el servidor responde directamente.



Arquitectura de tres capas

eso significa que la arquitectura generalmente está compartida por un cliente que como hablamos más arriba es el que solicita los recursos equipado con una interfaz de usuario o mediante un navegador web.

Arquitectura N capas

Los servidores dos y tres realizaron una tarea específica por lo tanto un servidor web puede usar los servicios de otros servidores para poder proporcionar su propio servicio.



VENTAJAS

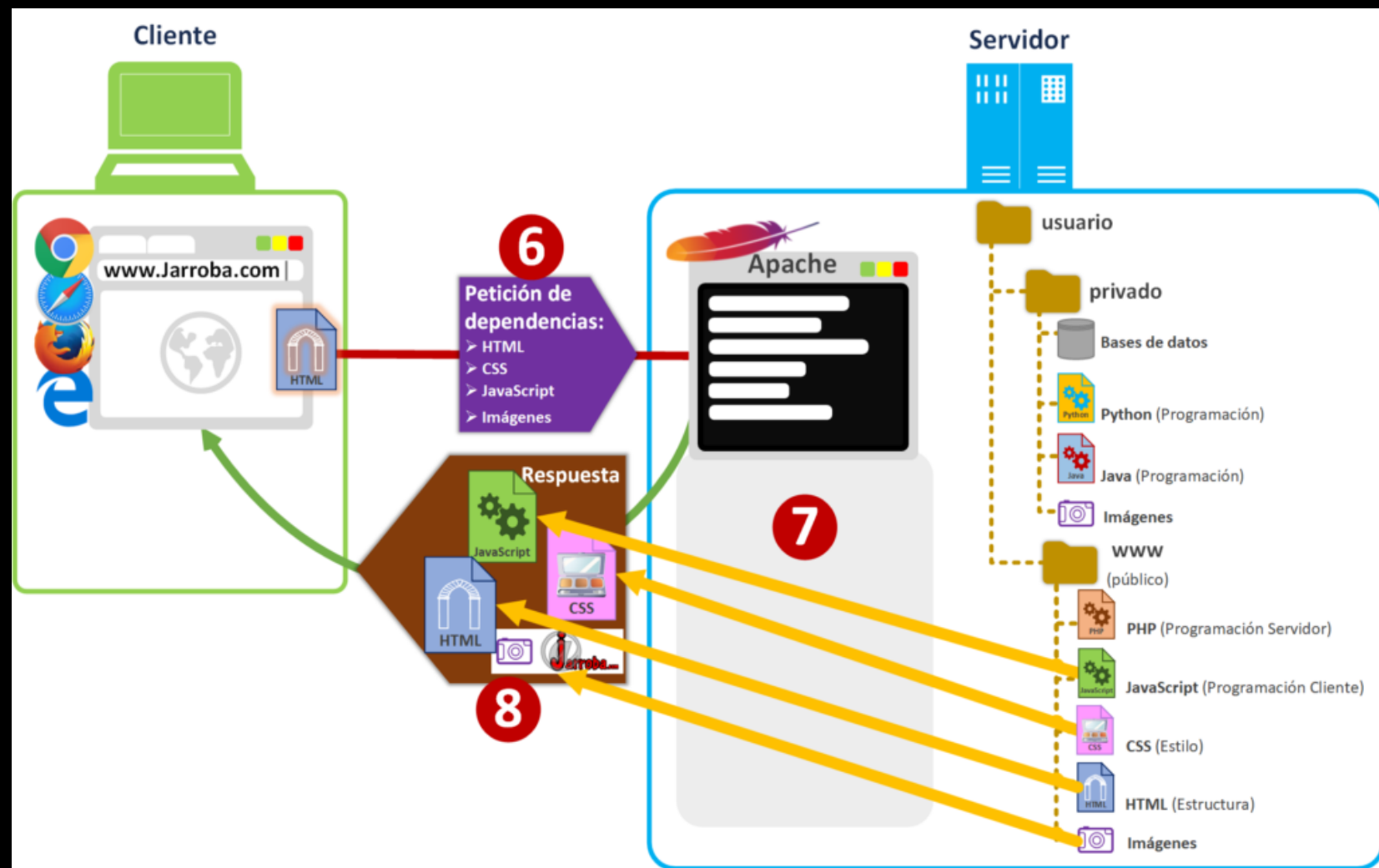
- ❑ La **distribución** de datos es **directa**.
- ❑ Permite el **uso efectivo de Sistema** de red. Generalmente requiere hardware barato.
- ❑ Es fácil **añadir** nuevos **servidores** o **actualizar** los existentes.
- ❑ Facilita la **integración** entre diferentes sistemas y comparte información, mediante una interfaz más amigable para el **usuario**.
- ❑ Al favorecer el uso de las interfaz de gráficas interactivas, los sistemas construidos bajo este esquema tienen una mayor interacción con el **usuario**.
- ❑ La estructura modular facilita de más la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional.



DESVENTAJAS

- El modelo **no comparte datos** con los diferentes subsistemas empleados en la organización. El **intercambio** de datos puede ser ineficiente.
- **Todo o nada**: Si el servidor **se cae**, todos los clientes quedarán totalmente **inoperables**.
- Si se intercambian grandes volúmenes de datos, quien más **sufrirá** será el **rendimiento**.
- Es más **difícil** realizar datamining o inteligencia de negocio.
- Es difícil **analizar** un error, pues los clientes están distribuidos en diferentes máquinas, incluso, equipos a los cuales no tenemos acceso, lo que hace complicado recopilar la traza del error.
- **No existen registros centrales** de nombres y servicios, esto hace difícil encontrar los servidores y servicios disponibles.

EJEMPLO



CONCLUSIÓN

A pesar de que el estilo **Cliente-Servidor** no es muy popular entre los nuevos desarrolladores, la realidad es que sigue siendo parte fundamental en muchas de las arquitecturas de hoy en día, solo basta decir que todo el internet está basado en Cliente-Servidor, sin embargo, no es común que como programadores o arquitectos nos encontremos ante problemáticas que requieran implementar un Cliente-Servidor.

GRACIAS POR SU
ATENCIÓN 😊