

# Arquitecturas Cliente/Servidor

Unidad I. Fundamentos

Tecnologías para Desarrollos en Internet

---

Jesús Iván Saavedra Martínez

Facultad de Ciencias, UNAM

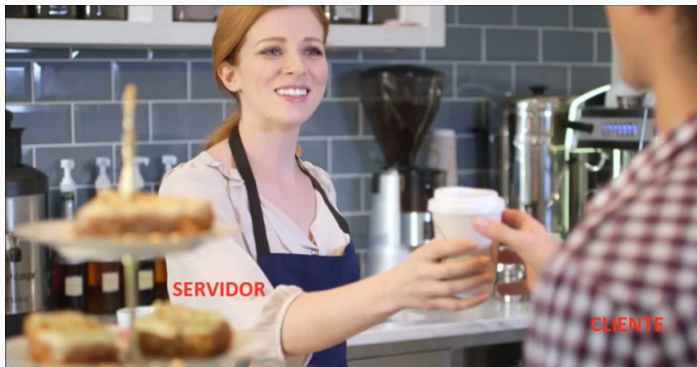
# Tabla de contenido

1. ¿Qué es la arquitectura Cliente/Servidor?
2. Comunicación entre el cliente y el servidor
3. Arquitectura cliente/servidor de n-capas
4. Ventajas y desventajas de la arquitectura Cliente/Servidor

## **¿Qué es la arquitectura Cliente/Servidor?**

---

# Analogía

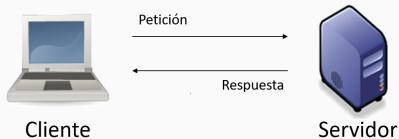


- Cliente: **solicita** un producto o servicio
- Servidor: **da respuesta** a la solicitud del cliente

# La arquitectura Cliente/Servidor

Cliente / servidor es un término genérico para cualquier arquitectura de aplicación que divide el procesamiento entre dos o más procesos.

- Un *proceso* es un programa que se encuentra en ejecución en algún equipo de cómputo.



## Cliente

- inicia solicitudes o peticiones
- tiene un papel activo en la comunicación
- espera y recibe las respuestas del servidor
- por lo general, puede conectarse a varios servidores a la vez
- normalmente interactúa directamente con los usuarios finales



Cliente

## Servidor

- está a la espera de que lleguen solicitudes de los clientes
- desempeña un papel pasivo en la comunicación
- tras la recepción de una solicitud, la procesa y envía la respuesta al cliente
- por lo general, acepta las conexiones de un gran número de clientes



Servidor

# Comunicación entre el cliente y el servidor

---



Comunicación entre procesos:

- Cuando dos procesos están en la misma computadora pueden usar mecanismos de comunicación que se encuentren definidos por parte del sistema operativo.
- Cuando dos procesos se encuentran en computadoras distintas **necesitan de mensajes para comunicarse.**

# Estructura de los mensajes

Para cada aplicación se debe definir la estructura de los mensajes, el significado de cada campo, el orden en que se intercambian, etc.

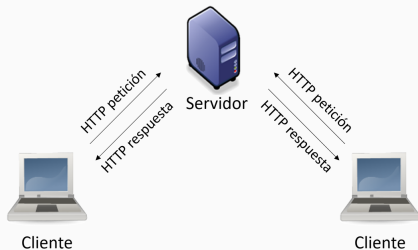
- Tipos de mensajes que se intercambian (solicitud, respuesta)
- Sintaxis de los mensajes, campos permitidos, orden, etc.
- Semántica de los campos en los mensajes. Significado de la información que aparece en cada campo.
- Reglas para determinar en qué momento se pueden enviar y responder los mensajes



# Protocolo HTTP

El Protocolo de transferencia de hipertexto (HTTP) permite la transferencia de información en la *www*. Desarrollado por W3C e IETF.

- **Define la estructura de los mensajes** que son intercambiados entre el cliente y el servidor
- Se apoya del protocolo TCP para envío de mensajes
- Independiente del sistema operativo.
- Orientado a conexión



# Métodos de petición (verbos) HTTP

HTTP define un conjunto de métodos (verbos) de petición para indicar la acción que se desea realizar para un recurso determinado.

Algunos de los métodos más utilizados se muestran en la Tabla 1, que corresponden a un CRUD.

**Tabla 1:** Verbos HTTP más frecuentes

| Verbo  | Descripción                          |
|--------|--------------------------------------|
| GET    | recupera elementos del recurso       |
| POST   | crea un nuevo elemento en el recurso |
| PUT    | reemplaza un elemento del recurso    |
| DELETE | elimina un elemento del recurso      |

# Arquitectura cliente/servidor de n-capas

---

# Arquitectura de 3 capas

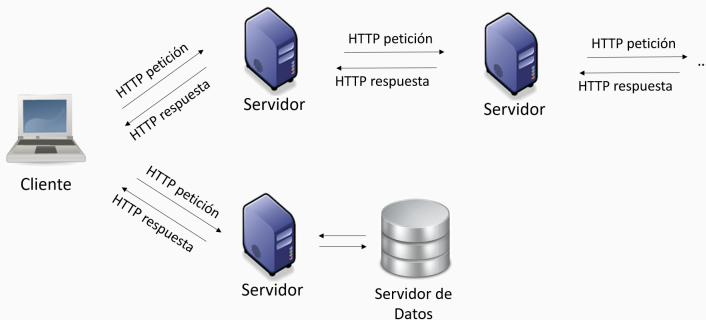
La forma más simple que adopta una arquitectura cliente / servidor se llama arquitectura de 2 capas (niveles). En la arquitectura de 3 capas existe un nivel intermediario, donde la arquitectura está comúnmente compartida por:

1. un cliente que solicita los recursos, generalmente equipado con una interfaz de usuario
2. el servidor de aplicaciones, que proporciona los recursos solicitados pero requiere de otro servidor
3. el servidor de datos que proporciona al servidor de aplicaciones los datos que requiere



# Arquitectura de n capas

- cada servidor en los niveles intermedios realizan una tarea especializada (servicio)
- un servidor puede utilizar los recursos de otro para proporcionar su propio servicio
- la arquitectura de 3 capas es potencialmente una arquitectura de n capas



## **Ventajas y desventajas de la arquitectura Cliente/Servidor**

---



- Control centralizado: los accesos, recursos y la integridad de los datos son controlados por el servidor
- Integración y mantenimiento: es posible actualizar, reemplazar, reparar o incluso migrar un servidor
- Escalabilidad: se puede aumentar la capacidad de la infraestructura de clientes y servidores, así añadir nuevos nodos a la red

- Congestión de tráfico: cuando una gran cantidad de clientes envían peticiones al mismo servidor generan congestiones en la red.
- Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- Comúnmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Esto implica costos el costo.

- I. R. Govett, R. Vt, Client/server architecture supporting concurrent servers within a server with a transaction manager providing server/connection decoupling, International Business Machines Corporation, Armonk, N.Y. Jun 1998.
- G. Reese, Database Programming with JDBC & Java, Second Edition, O'Reilly Media, Inc., August 2000.
- J. F. Kurose, K. W. Ross, ComputerNetworking: A Top-Down Approach, Pearson Education, Inc., Sixth Edition.