

## 3 Arquitectura Cliente/Servidor

---

El termino cliente/servidor se refiere normalmente a los sistemas que dividen el procesamiento entre uno o más clientes y un servidor central conectados a través de una red. En un sistema cliente/servidor típico, el cliente maneja la interfaz de usuario, incluyendo la entrada de datos, la consulta de datos y la lógica de la presentación en pantalla. El servidor almacena los datos y provee el acceso a los datos y las funciones de la administración de la base de datos. De alguna manera, la lógica de la aplicación está dividida entre el servidor y los clientes.

Para completar la petición de un cliente, el servidor puede necesitar contactar otros servidores por soporte de datos o procesamiento, pero éste proceso es transparente para los clientes.

A medida que las redes de gran escala incrementan su poder, los sistemas cliente/servidor se vuelven más eficientes económicamente. Hoy en día la arquitectura cliente/servidor sigue siendo un diseño de sistemas popular, usando protocolos de Internet y modelos de redes como los descritos anteriormente. A medida que los negocios forman nuevas alianzas entre clientes y proveedores, el concepto de cliente/servidor se expande para incluir clientes y servidores fuera de la organización. La arquitectura orientada al servicio ("SOA" del inglés "Service-Oriented Architecture") es un ejemplo de un sistema conectado donde un servicio puede ser simultáneamente un cliente y un servidor y puede existir fuera de los límites de la corporación.

La computación en la nube (en inglés "Cloud Computing") es vista por algunos como un concepto totalmente nuevo. Otros la ven como la culminación de la arquitectura cliente/servidor, donde la computación basada en Internet se vuelve la parte "servidor" que administra las tareas de procesamiento, mientras que la misma Internet se convierte en la plataforma que reemplaza las redes tradicionales.

### El Rol del Cliente

La relación cliente/servidor debe especificar cómo se dividirá el procesamiento entre el cliente y el servidor. Mientras que el diseño de "cliente pesado" (en inglés "Rich/Heavy/Fat/Thick Client") ubica toda o la mayoría de la lógica de procesamiento del lado del cliente, el diseño de "cliente liviano" (en inglés "Thin Client") ubica toda o la mayoría de la lógica de procesamiento del lado del servidor.

### Niveles de Cliente/Servidor

A los primeros diseños de cliente/servidor se los llamó "diseño de dos niveles" (en inglés "two-tier design"). En un diseño de dos niveles, la interfaz de usuario reside en el cliente, todos los datos residen en el servidor y la lógica de aplicación puede ejecutarse tanto en el servidor como en el cliente o dividirse entre ambos.

Existe otra forma de diseño de cliente/servidor, llamado "diseño de tres niveles" (en inglés "three-tier design"), que se ha vuelto más popular. En un diseño de tres niveles la interfaz de usuario se ejecuta en el cliente y los datos se almacenan en el servidor, de la misma forma que en el diseño de dos niveles. El diseño de tres niveles también está compuesto de una capa intermedia, ubicada entre cliente y servidor, que procesa las peticiones del cliente y las traduce en comandos que el servidor puede entender y llevar a cabo para acceder a los datos. La capa intermedia puede considerarse como un servidor de aplicación, ya que provee la lógica de la aplicación, o la lógica de negocios, requerida por el sistema. A los diseños de tres niveles también se los llama "diseños de n-niveles" (en inglés "n-tier designs"), para indicar que algunos diseños usan más de una capa intermedia.

## Software Intermedio

En un sistema multiniveles, al software especial que permite que los niveles se comuniquen y que los datos pasen un lado al otro, se lo llama "software intermedio" (en inglés "middleware"). Al middleware también se lo conoce como "software pegamento" (en inglés "glueware"), porque se usa para conectar dos o más componentes de software en una arquitectura de sistema federado.

## Problemas de Costo-Beneficio

Para dar soporte a los requerimientos de negocios, los sistemas informáticos necesitan ser escalables, potentes y flexibles. Para la mayoría de las compañías, los sistemas cliente/servidor ofrecen la mejor combinación de características para cumplir con éstas necesidades. Ya sea que un negocio se expande o contrae, los sistemas cliente/servidor le permiten a la firma escalar el sistema dentro de un entorno que está en constante evolución.

La computación cliente/servidor también le permite a las compañías el transferir aplicaciones desde una plataforma cara como la de los mainframes a una más económica como la de los clientes, a veces moviendo a la nube todo lo relacionado con los procesos pesados (que demandan mayor poder computacional).

## Problemas de Rendimiento

En contraste con los sistemas centralizados, el diseño cliente/servidor separa la aplicación y los datos. Los clientes conectados envían las peticiones de datos al servidor, el cual responde enviando los datos a los clientes. Cuando el número de clientes y la demanda se incrementa más allá de cierto nivel, la capacidad de la red se convierte en un cuello de botella y el rendimiento del sistema decae considerablemente.