



Arquitecturas: Cliente-Servidor y de N capas

Facultad de Ciencias de la Computación

Juan Carlos Conde Ramírez

Web Technologies

Contenido

1 Introducción

2 Distribución

3 Modelos

4 Niveles

Contenido

1 Introducción

2 Distribución

3 Modelos

4 Niveles

Características Básicas, I

La arquitectura de red Cliente-Servidor es aquella en la que cada computadora (proceso) en la red "pide" o "sirve" recursos. Las aplicaciones Web son un tipo especial de aplicaciones tipo "Cliente-Servidor"

Otras arquitecturas...

- ▶ Otro tipo de arquitectura es la P2P (*Peer-to-Peer*) ó de igual a igual, en la que cada computadora de la red posee responsabilidades equivalentes.
- ▶ Aquí una misma computadora puede ser cliente y servidor simultáneamente, por lo que se establece sólo una separación lógica según las funciones que realiza.

Características Básicas, II

Normalmente, los servidores son computadoras potentes dedicados a gestionar:

- ▶ unidades de disco (servidor de archivos),
- ▶ impresoras (servidor de impresoras),
- ▶ tráfico de red (servidor de red),
- ▶ datos (servidor de bases de datos) o incluso
- ▶ aplicaciones (servidor de aplicaciones).

Mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.

Características Básicas, III

Dentro de los clientes se suelen distinguir dos clases:

1. **Clientes inteligentes o enriquecidos** (*rich client*). Son computadoras completas, con todo el hardware y software necesarios para poder funcionar de forma independiente.
2. **Clientes tontos o ligeros** (*thin client*). Son terminales que no pueden funcionar de forma independiente, ya que necesitan de un servidor para ser operativos.

Características Básicas, IV

Esta arquitectura implica la existencia de una **relación entre procesos** que solicitan servicios (*request*) y procesos que responden a estos servicios (*response*).

Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos.

La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio, permitiendo la creación de **aplicaciones distribuidas**, es decir, situar cada función en la plataforma más adecuada para su ejecución.

Contenido

1 Introducción

2 Distribución

3 Modelos

4 Niveles

Características Avanzadas, I

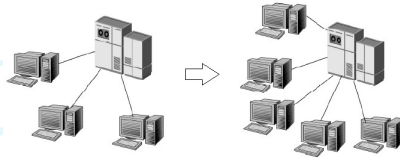
Dado que las redes de computadoras permiten que múltiples procesadores puedan ejecutar partes de una misma aplicación, logrando concurrencia¹ de procesos, existen otro tipo de ventajas del uso de una arquitectura como la Cliente-Servidor:

- ▶ Posibilita la migración aplicaciones de un procesador a otro con modificaciones mínimas en los programas.
- ▶ Posibilita el acceso a los datos independientemente de donde se encuentre el usuario.
- ▶ La aplicación se vuelve escalable (ampliación horizontal o vertical).

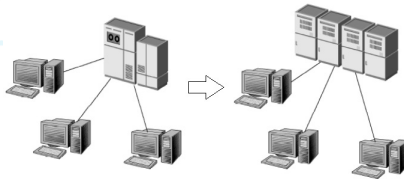
¹Procesos que comparten datos, y por tanto sincronizan su acceso a los datos de modo que las actualizaciones de los mismos no se pierdan o corrompan como resultado de los accesos.

Características Avanzadas, II

Escalabilidad horizontal: capacidad de añadir o suprimir estaciones de trabajo (clientes) que hagan uso de la aplicación, sin que afecte sustancialmente al rendimiento general.

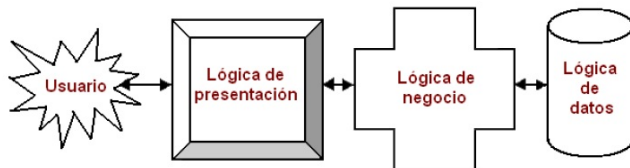


Escalabilidad vertical: capacidad de migrar hacia servidores de mayor capacidad o velocidad, o de un tipo distinto de arquitectura sin que afecte a los clientes.



Separación de Funciones, I

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles, tal como se muestra a continuación:



Separación de Funciones, II



Definición

Se encarga de la entrada y salida de la aplicación con el usuario.

Tareas:

- ▶ Obtener información del usuario, enviar la información del usuario a la lógica de negocio para su procesamiento,
- ▶ recibir los resultados del procesamiento de la lógica de negocio y
- ▶ presentar estos resultados al usuario.

Separación de Funciones, III



Definición

Se encarga de gestionar los datos a nivel de procesamiento. Actúa de puente entre el usuario y los datos

Tareas:

- ▶ Recibir la entrada del nivel de presentación,
- ▶ interactuar con la lógica de datos para ejecutar las reglas de negocio (*business rules*) que tiene que cumplir la aplicación (facturación, cálculo de nóminas, control de inventario, etc.) y
- ▶ enviar el resultado del procesamiento al nivel de presentación.

Separación de Funciones, IV



Definición

Se encarga de gestionar los datos a nivel de almacenamiento.

Tareas:

- ▶ Almacenar los datos,
- ▶ recuperar los datos,
- ▶ mantener los datos y
- ▶ asegurar la integridad de los datos.

Contenido

1 Introducción

2 Distribución

3 Modelos

4 Niveles

Modelos de Distribución, I

Si un sistema distribuido se diseña correctamente, los tres niveles anteriores pueden distribuirse y redistribuirse independientemente sin afectar el funcionamiento de la aplicación. Según como se distribuyan podemos considerar 3 modelos:

1. **Presentación** distribuida,
2. **Aplicación** distribuida y
3. **Datos** distribuidos.

Modelos de Distribución, II

Presentación distribuida:

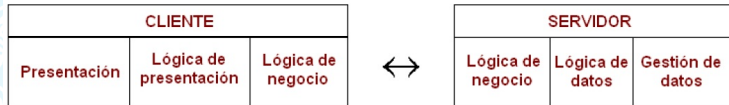
- ▶ El cliente sólo mantiene la presentación, el resto de la aplicación se ejecuta remotamente.
- ▶ En su forma más simple, es una interfaz gráfica de usuario a la que se le pueden acoplar validación de datos para evitar la validación en el servidor.



Modelos de Distribución, III

Aplicación distribuida:

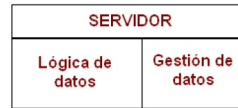
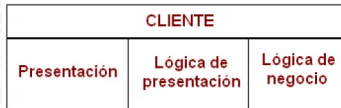
- ▶ Proporciona máxima flexibilidad; permite tanto al servidor como al cliente mantener la lógica de negocio.
- ▶ Tanto Cliente como Servidor realizan las funciones que le sean más propias; ya sea por mejora de la organización o mejora en el rendimiento del sistema.



Modelos de Distribución, IV

Datos distribuidos:

- ▶ Los datos son los que se distribuyen, por lo que la lógica de datos es lo que queda separada del resto de la aplicación.
- ▶ Se puede dar de dos formas: **archivos** distribuidos o **bases de datos** distribuidas.



Contenido

1 Introducción

2 Distribución

3 Modelos

4 Niveles

Arquitectura de 2 y 3 Niveles, I

La diferencia entre una arquitecturas de 2 y una de 3 niveles (*tiers*) estriba en la forma de **distribución de la aplicación entre el Cliente y el Servidor**.

Cuando se habla de aplicaciones de 2 niveles se trata de una aplicación donde el Cliente mantiene la *lógica de presentación* y de *negocio*, y el Servidor únicamente gestiona los *datos*.

Suelen ser aplicaciones cerradas que supeditan las solicitudes de los procesos del Cliente al **gestor de bases de datos** que se está usando.

Arquitectura de 2 y 3 Niveles, II

En una arquitectura de 3 niveles la *lógica de presentación*, la *lógica de negocio* y la *lógica de datos* están separadas.

Mientras la lógica de presentación se ejecutará normalmente en la estación cliente, la lógica de negocio y la de datos pueden estar repartidas entre distintas estaciones servidor.

Suelen existir dos servidores: uno que contiene la lógica de negocio y otro que contiene lógica de datos.



Arquitectura de 2 y 3 Niveles, III

El objetivo de aumentar el No. de niveles, **en una aplicación distribuida**, es lograr una mayor independencia entre un nivel y otro, lo que facilita la portabilidad en entornos heterogéneos y la escalabilidad en caso de incorporación de nuevos clientes.

Sistemas Cliente-Servidor, I

Un sistema Cliente-Servidor suele presentar las siguientes características:

1. Una parte cliente también llamada **front-end** que interactúa con el usuario (interfaz) y la parte servidor o **back-end** que interactúa con los recursos compartidos (bases de datos, impresoras, módems).
2. Las partes cliente y servidor tienen diferentes necesidades de recursos a la hora de ejecutarse: velocidad de procesador, memoria y capacidad disco(s) duro(s), dispositivos de entrada/salida, etc.

Sistemas Cliente-Servidor, II

3. El entorno suele ser heterogéneo y multi-vendedor (vendor) (hardware y software de distintos fabricantes). El Hardware y sistema operativo del cliente y el servidor suelen diferir.
4. El cliente y el servidor se suelen comunicar a través de una Application Program Interface (API) y Remote Procedure Call (RPC) conocidas.
5. Normalmente la parte cliente se implementa permitiendo la introducción de datos a través de teclado, ratón, lápiz óptico, etc.

Comentarios...

Juan Carlos Conde R.
juanc.conde@cs.buap.mx