

# SISTEMAS DISTRIBUIDOS DE REDES

## 1. SISTEMAS DISTRIBUIDOS

### *Introducción y generalidades*

*La computación desde sus inicios ha sufrido muchos cambios, desde los grandes equipos que permitían realizar tareas de forma limitada y de uso un tanto exclusivo de grandes organizaciones, hasta los actuales ya sean personales o portátiles que tienen las mismas e incluso mayores capacidades que los primeros y que están cada vez más introducidos en el quehacer cotidiano de una persona.*

*Los mayores cambios se dieron en las décadas de los setenta, atribuidos principalmente a dos causas:*

*\* El desarrollo de los microprocesadores, que permitieron reducir en tamaño y costo a los ordenadores y aumentar en gran medida las capacidades de los mismos y su acceso a más personas.*

*\* El desarrollo de las redes de área local y de las comunicaciones que permitieron conectar ordenadores con posibilidad de transferencia de datos a alta velocidad.*

*Es en este contexto que aparece el concepto de "Sistemas Distribuidos" que se ha popularizado tanto en la actualidad y que tiene como ámbito de estudio las redes como por ejemplo: Internet, redes de teléfonos móviles, redes corporativas, redes de empresas, etc.*

### **Modelos de desarrollo**

Definición de Sistemas Distribuidos:

"Sistemas cuyos componentes hardware y software, que se encuentran en equipos conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. La comunicación se establece mediante el uso de un protocolo prefijado para un esquema cliente-servidor".

Características:

- **Concurrencia.**- Esta característica en los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que se encuentran interactuando en una red.
- **Carencia de reloj global.**- Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no requieren de una temporización general, esta se encuentra distribuida en los componentes.
- **Fallos independientes de los componentes.**- Cada componente del sistema puede fallar independientemente, con lo cual los demás pueden continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

## **Origen y evolución del modelo cliente y servidor**

Origen:

La Computación Cliente Servidor.- Este modelo nace de la evolución de los sistemas distribuidos y es el que predomina en la actualidad, ya que permite descentralizar el procesamiento y recursos, sobre todo, en lo correspondiente a los servicios y la visualización de la Interfaz Gráfica del Usuario. Esto hace que ciertos servidores estén dedicados solo a una aplicación determinada y por lo tanto su ejecución es más eficiente.

Definición:

Sistema donde el cliente es una máquina que solicita un determinado servicio y se denomina servidor a la máquina que lo proporciona. Los servicios pueden ser:

- Ejecución de un determinado programa.
- Acceso a un determinado banco de información.
- Acceso a un dispositivo de hardware.
- Es un elemento primordial, la presencia de un medio físico de comunicación entre las máquinas, y dependerá de la naturaleza de este medio la viabilidad del sistema.

Categorías de Servidores:

A continuación se presenta una lista de los servidores más comunes:

Servidores de archivos.- Proporciona archivos para clientes. Si los archivos no fueran tan grandes y los usuarios que comparten esos archivos no fueran muchos, esto sería una gran opción de almacenamiento y procesamiento de archivos. El cliente solicita los archivos y el servidor los ubica y se los envía.

Servidores de Base de Datos.- Son los que almacenan gran cantidad de datos estructurados, se diferencian de los de archivos pues la información que se envía está ya resumida en la base de datos. Ejemplo: El Cliente hace una consulta, el servidor recibe esa consulta (SQL) y extrae solo la información pertinente y envía esa respuesta al cliente.

Servidores de Software de Grupo.- El software de grupo es aquel, que permite organizar el trabajo de un grupo. El servidor gestiona los datos que dan soporte a estas tareas. Por ejemplo: almacenar las listas de correo electrónico. El Cliente puede indicarle, que se ha terminado una tarea y el servidor se lo envía al resto del grupo.

Servidores WEB.- Son los que guardan y proporcionan Páginas HTML. El cliente desde un browser o link hace un llamado de la página y el servidor recibe el mensaje y envía la página correspondiente.

Servidores de correo.- Gestiona el envío y recepción de correo de un grupo de usuarios (el servidor no necesita ser muy potente). El servidor solo debe utilizar un protocolo de correo (SMTP).

Servidor de objetos.- Permite almacenar objetos que pueden ser activados a distancia. Los clientes pueden ser capaces de activar los objetos que se encuentran en el servidor.

Servidores de impresión.- Gestionan las solicitudes de impresión de los clientes. El cliente envía la solicitud de impresión, el servidor recibe la solicitud y la ubica en la cola de impresión, ordena a la impresora que lleve a cabo las operaciones y luego avisa a la computadora cliente que ya acabo su respectiva impresión.

Servidores de aplicación.- Se dedica a una única aplicación. Es básicamente una aplicación a la que pueden acceder los clientes.

Componentes de Software:

Se distinguen tres componentes básicos de software:

- **Presentación.-** Tiene que ver con la presentación al usuario de un conjunto de objetos visuales y llevar a cabo el procesamiento de los datos producidos por el mismo y los entregados por el servidor.
- **Lógica de aplicación.-** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos.-** Esta compuesta de los archivos que contienen los datos de la aplicación.

### **Evolución:**

**Procesamiento central (Host).**- Uno de los primeros modelos de equipos interconectados, llamados centralizados, donde todo el procesamiento de la organización se lleva a cabo en una sola computadora, normalmente un Mainframe, y los usuarios empleaban sencillos equipos personales.

Los problemas de este modelo son:

Cuando la carga de procesamiento aumenta se tiene que cambiar el incrementar las capacidades en hardware del Mainframe, lo cual es más costoso que añadir más equipos personales clientes o servidores que aumenten la capacidad de desempeño.

El otro problema que surge son las modernas interfases gráficas de usuario, las cuales pueden conllevar a un gran aumento de tráfico en los medios de comunicación y por consiguiente generar fallas operativas.

A pesar de los problemas que presenta este procesamiento, actualmente sigue siendo utilizado en la estructura operativa de algunas aplicaciones que no requieren un esquema tipo cliente-servidor.

### **Evolución:**

**Grupo de Servidores.**- Otro modelo que entró a competir con el anterior, también un tanto centralizado, es el formado por un grupo de equipos actuando como servidores, normalmente de archivos o de impresión, poco inteligentes para un número de Minicomputadores que hacen el procesamiento conectados a una red de área local.

Los **problemas** de este **modelo** son:

Puede generarse una saturación de los medios de comunicación entre los servidores poco inteligentes y los minicomputadores, por ejemplo cuando se solicitan archivos grandes por varios clientes a la vez, podían disminuir en gran medida la velocidad de transmisión de la información.

**La Computación Cliente Servidor.**- Este modelo se menciona a mayor detalle en el siguiente subtema.

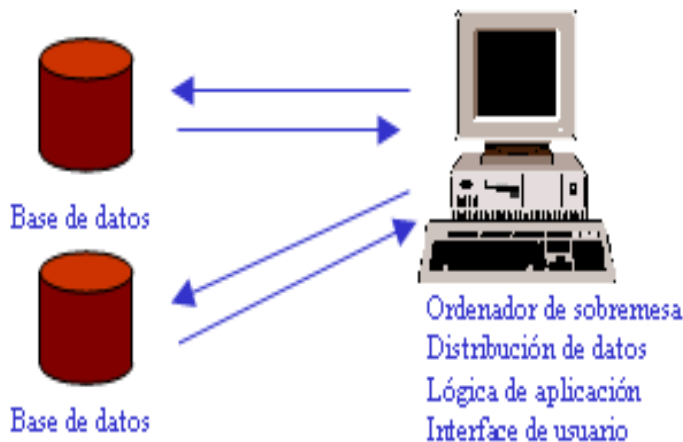
## Arquitecturas cliente y servidor basadas en dos y tres niveles de comunicaciones

A continuación se muestra las arquitecturas cliente-servidor más populares:

### CLIENTE/SERVIDOR DE DOS CAPAS

- **Arquitectura Cliente-Servidor de Dos Capas.-** Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:
  - Cuando se requiere poco procesamiento de datos.
  - Cuando se tiene una base de datos centralizada en uno o varios servidores.
  - Cuando la base de datos es relativamente estática.
  - Cuando se requiere un mantenimiento mínimo.

Un ejemplo es, el cliente realiza la interacción con el usuario y ejecuta la aplicación, así como la solicitud de información a la base de datos. Para este esquema se debe conocer la topología de la red, así como la disposición y ubicación de los datos. Se delega parte de la gestión de la base de datos al cliente.



### CLIENTE/SERVIDOR DE TRES CAPAS

- **Arquitectura Cliente-Servidor de Tres Capas-** Consiste en una capa de presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:
  - Cuando se requiere mucho procesamiento de datos en la aplicación.
  - En aplicaciones donde la funcionalidad este en constante cambio.
  - Cuando los procesos no están relativamente muy relacionados con los datos.
  - Cuando se requiere aislar la tecnología de la base de datos para que sea fácil de cambiar.
  - Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.

- Es muy adecuada para utilizarla en tecnología orientada a objetos.

Un ejemplo, el cliente se encarga de la interacción con el usuario, el servidor de la lógica de aplicación y la base de datos puede estar en otro servidor.



## BIBLIOGRAFIA

- [http://fmc.axarnet.es/redes/tema\\_04.htm](http://fmc.axarnet.es/redes/tema_04.htm) (Sistemas Operativos).
- <http://www.fortunecity.es/sopa/chinchulines/812/informacion/noscs.htm> (Sistemas Operativos).
- <http://dmi.uib.es/~bbuades/sistdistr/sld007.htm> (Sistemas Distribuidos).
- <http://members.fortunecity.es/lrmdl/SO7.htm#VSDRC> (Sistemas Distribuidos).
- <http://sacbeob.8m.com/tutoriales/bddistribuidas/> (Base de Datos Distribuidas).
- <http://pdf.rincondelvago.com/bases-de-datos-distribuidas.html> (Base de Datos Distribuidas)
- [http://www-lt.ls.fi.upm.es/sistemas\\_dist/Introduccion.pdf](http://www-lt.ls.fi.upm.es/sistemas_dist/Introduccion.pdf) (Sistemas Distribuidos).
- [http://www.dia.eui.upm.es/cgi-bin/asiqfram.pl?cual=sis\\_dis&nombre=Sistemas-Distribuidos](http://www.dia.eui.upm.es/cgi-bin/asiqfram.pl?cual=sis_dis&nombre=Sistemas-Distribuidos) (Sistemas Distribuidos: Aplicaciones).
- <http://www.fisica.uson.mx/carlos/WebServices/WSOverview.htm> (Objetos Distribuidos).
- <http://di002.edv.uniovi.es/~lourdes/publicaciones/bt99.pdf> (Objetos Distribuidos).
- <http://www.iespana.es/darkstar/secciones/informatica/redes/tcpip/capas.html> (Protocolos).