1.3.ORDINADORS CLIENTS I ORDINADORS SERVIDORS.

La **arquitectura cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes **computadoras** aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se este utilizando en una red mixta.

Características

En la arquitectura C/S el **remitente de una solicitud** es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro** o **amo**).
- 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 mediante una interfaz gráfica de usuario.
- Al contratar un servicio de redes , se tiene que tener en la velocidad de conexión que le otorga al cliente y el tipo de cable que utiliza , por ejemplo : cable de cobre ronda entre 1 ms y 50 ms.

Al **receptor de la solicitud** enviada por el cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Comparación de la arquitectura C/S con otras arquitecturas de red

Comparación con las redes de pares

Las **redes de pares**, también conocidas como redes **par-a-par** o **peer-to-peer** (abreviado con las siglas **P2P**M) son otro tipo de arquitectura de red.

Comparación con la arquitectura Cliente-Cola-Cliente

Si bien la clásica arquitectura C/S requiere uno de los puntos terminales de comunicación para actuar como un servidor, que puede ser algo más difícil de aplicar, la arquitectura Cliente-Cola-Cliente habilita a todos los nodos para actuar como clientes simples, mientras que el servidor actúa como una cola que va capturando las peticiones de los clientes (un proceso que debe pasar sus peticiones a otro, lo hace a través de una cola, por ejemplo, una consulta a una base de datos, entonces, el segundo proceso conecta con la base de datos, elabora la petición, la pasa a la base de datos, etc.). Esta arquitectura permite simplificar en gran medida la implementación de software. La arquitectura P2P originalmente se basó en el concepto "Cliente-Cola-Cliente".

Arquitecturas multi-capas

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o **dos capas**.

Algunas redes disponen de tres tipos de nodos:

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una arquitectura de tres-capas.

- Ventajas de las arquitecturas n-capas:

  La ventaja fundamental de una arquitectura **n-capas** comparado con una arquitectura de dos niveles (o una tres-capas con una de dos niveles) es que separa hacia fuera el proceso, eso ocurre para mejorar el balance la carga en los diversos servidores; es más escalable.

- Desventajas de las arquitecturas de la n-capas:

  - Pone más carga en la red, debido a una mayor cantidad de tráfico de la red.
  - Es mucho más difícil programar y probar el software que en arquitectura de dos niveles porque tienen que comunicarse más dispositivos para terminar la transacción de un usuario.

Ventajas

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P)..
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

Desventajas

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuanto más nodos hay, mejor es el ancho de banda que se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está *caído*, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

Dirección

Los métodos de dirección en ambientes del servidor de cliente se pueden describir como sigue:

- Dirección del proceso de la máquina: la dirección se divide como proceso@máquina. Por lo tanto 56@453 indicaría el proceso 56 en la computadora 453.
- Servidor de nombres: los servidores de nombres tienen un índice de todos los nombres y direcciones de servidores en el dominio relevante.
- Localización de Paquetes: Los mensajes de difusión se envían a todas las computadoras en el sistema distribuido para determinar la dirección de la computadora de la destinación.
- Comerciante: Un comerciante es un sistema que pone en un índice todos los servicios disponibles en un sistema distribuido. Una computadora que requiere un servicio particular comprobará con el servicio que negocia para saber si existe la dirección de una computadora que proporciona tal servicio.

Ejemplos

La mayoría de los servicios de Internet son tipo de cliente-servidor. La acción de visitar un sitio web requiere una arquitectura cliente-servidor, ya que el servidor web sirve las páginas web al navegador (al cliente). Al leer este artículo en Wikipedia , la

computadora y el navegador web del usuario serían considerados un cliente; y las computadoras, las bases de datos, y los usos que componen Wikipedia serían considerados el servidor. Cuando el navegador web del usuario solicita un artículo particular de Wikipedia, el servidor de Wikipedia recopila toda la información a mostrar en la base de datos de Wikipedia, la articula en una página web, y la envía de nuevo al navegador web del cliente.

Otro ejemplo podría ser el funcionamiento de un juego online. Si existen dos servidores de juego, cuando un usuario lo descarga y lo instala en su computadora pasa a ser un cliente. Si tres personasj juegan en un solo computador existirían dos servidores, un cliente y tres usuarios. Si cada usuario instala el juego en su propio ordenador existirían dos servidores, tres clientes y tres usuarios.

Cooperación cliente-servidor

Multiple Server

Para que un proceso se realice de la mejor manera, es preferible utilizar terminales distintos realizando la misma tarea, a centralizar los recursos y que con más hardware/software se realice la misma tarea. Con la ejecución de múltiples servidores el procesamiento es más rápido, el tiempo de respuesta es descentralizado y se incrementa la confiabilidad.

Cooperación de procesos paralelos

El mismo proceso se ejecuta simultáneamente (sistemas redundantes).

Cooperación de base de datos

Si se requiere de cierta información ya existente, por qué crearla de nuevo, simplemente interactúa y aprovecha la información ya creada.