

MÓDULO V

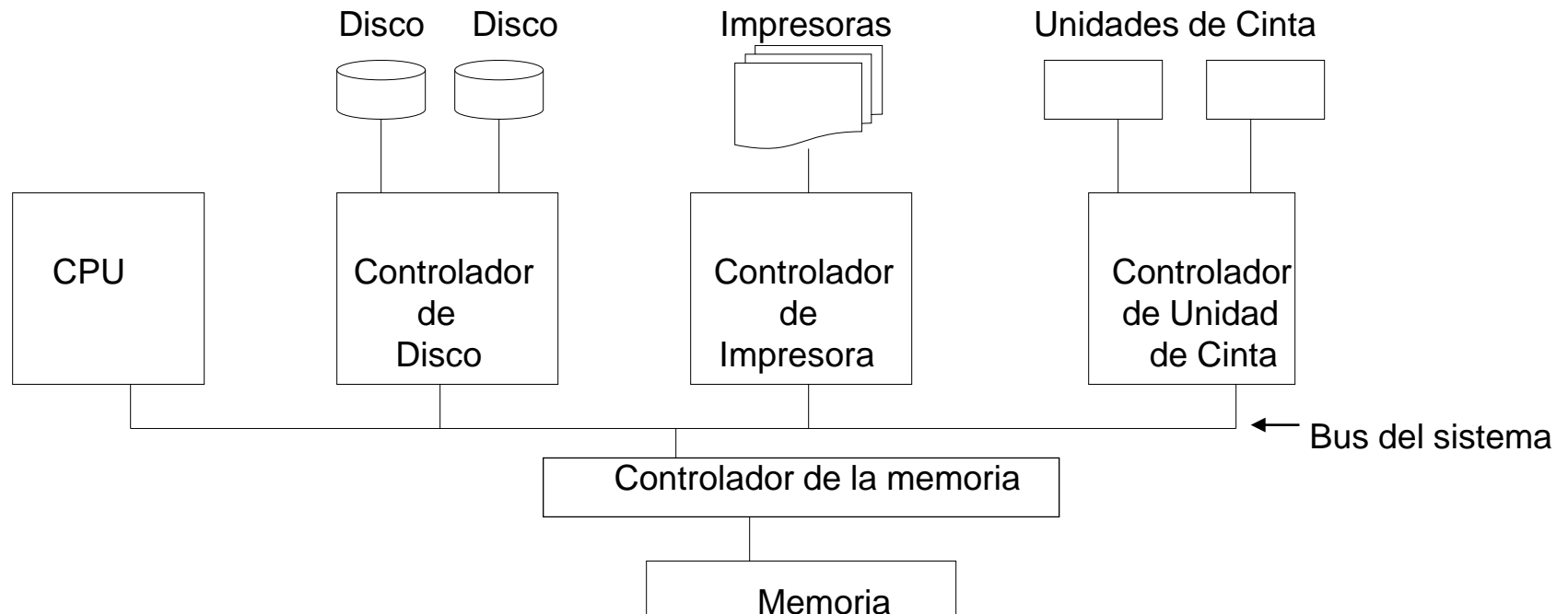
ARQUITECTURA CENTRALIZADA
ARQUITECTURA CLIENTE SERVIDOR
ARQUITECTURA DE 2 Y 3 CAPAS
TIPOS DE SERVIDORES:
 ITERATIVOS
 CONCURRENTES
 DE TRANSACCIÓN
 DE DATOS

Marcela Russo
Laboratorio IV

CARACTERISTICAS GENERALES DE LOS SGBD

➤ ARQUITECTURA CENTRALIZADA

- En la ARQUITECTURA CENTRALIZADA el SGBD está implementado en una sola plataforma (ordenador) desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos
- Los sistemas de BD centralizados se ejecutan en un único sistema informático, no interactúan con otra computadora.



- Una computadora de propósito general consiste de una o más CPU's y controladores para los dispositivos que se encuentren conectados a través de un bus de datos en común que proporciona el acceso a la memoria compartida
- Las CPU's poseen memorias cache locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos.
- Cada controlador se encarga de un tipo de dispositivo específico .
- Las CPU's y Controladores pueden ejecutarse concurrentemente compitiendo así por el acceso a la memoria.
- La Memoria Caché reduce la disputa por el acceso a la memoria principal, ya que la CPU necesita acceder a la memoria compartida menos cantidad de veces.
- Hay dos formas de utilizar las computadoras:
 - ✓ Sistemas monousuarios
 - ✓ Sistemas mutiusuarios

✓ **SISTEMAS MONOUSUARIOS**

- Computadoras personales y estaciones de trabajo.
- Se dispone de una sola CPU, con uno o dos discos físicos.
- El sistema operativo permite sólo un usuario.
- No poseen control de concurrencia.
- Las posibilidades de recupero son escasas y muy primitivas
- La mayoría no adminten SQL, proporcionan un lenguaje de consulta muy simple derivado del QBE (Query By Example)
- Las bases de datos que se ejecutan en estos sistemas disponen de multitarea, se ejecutan varios procesos a la vez en el mismo procesador, usando tiempo compartido, y de cara al usuario pareciese que los procesos se ejecutan en paralelo

✓ **SISTEMAS MULTIUSUARIOS**

- **Dispone de varias CPU's.**
- **El sistema operativo es multiusuario.**
- **Dan servicio a un gran número de usuarios que están conectados al sistema a través de terminales.**
- **Reciben el nombre de *SISTEMAS SERVIDORES*.**
- **Utilizan paralelismo disponiendo de unos pocos procesadores (normalmente entre 2 o 4) que comparten la misma memoria principal.**
- **En estas máquinas las bases de datos ejecutan las consultas en un único procesador, posibilitando la concurrencia de varias consultas.**
- **Permiten ejecutar un gran número de transacciones por segundo aumentando así la velocidad de respuesta, a pesar que cada transacción individualmente no se ejecuta más rápido.**
- **Estas máquinas tienen un gran número de procesadores y los sistemas de bases de datos intentan paralelizar las tareas simples.**

DISTANCIAMIENTO A LOS SISTEMAS CENTRALIZADOS – INICIOS DE LA ARQUITECTURA CLIENT-SERVIDOR

- Los sistemas se han ido alejando de la arquitectura centralizada debido al aumento de la velocidad y potencia de las computadoras personales y decremento en sus precios.
- Las terminales conectadas a un sistema central han sido suplantadas por computadoras personales.
- La interfaz de usuario gestionada directamente por el sistema central pasó a ser gestionada por computadoras personales.
- Los sistemas centralizados hoy actúan como sistemas servidores que satisfacen las peticiones generadas por los sistemas clientes.
- Esta arquitectura fué la antecesora a la arquitectura *Cliente - Servidor*
- La arquitectura *Cliente - Servidor* se la entiende como la extensión lógica de la programación modular, es decir la división de un programa grande en pequeños programas (llamados *módulos*) facilitando así el desarrollo y el mantenimiento (divide y vencerás).

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

- Por lo tanto motivaron el enfoque del procesamiento Cliente-Servidor:
 - ✓ El uso de recursos de cómputo remoto para realizar complejos procesos empresariales que consisten de una diversidad de subtareas.
 - ✓ Los clientes y servidores pueden estar ordenados a través de computadoras en red para dividir el trabajo complejo en unidades más manejables
 - ✓ El ordenamiento más simple es dividir el trabajo entre clientes que procesen en computadoras personales y un servidor que procese en una computadora separada

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

- Generalidades
- ✓ Los clientes acceden a los recursos vía aplicaciones (programas) para entregar la información vía algún protocolo de comunicación (por ejemplo TCP/IP).
- ✓ Así los sistemas Cliente/Servidor no están limitados solamente a aplicaciones de BD.
- ✓ Cualquier aplicación que contiene una interfaz de usuario (front-end, sección frontal o parte cliente) que se ejecute localmente en el cliente y un proceso que se ejecute en el servidor (back-end, sección posterior o sistema subyacente) está en formato de sistema Cliente/Servidor, o simplemente se puede decir que: un cliente es un programa que envía solicitudes a un servidor y un servidor procesa las solicitudes del cliente
- ✓ Conceptualmente las plataformas Cliente/Servidor son parte del concepto de **SISTEMAS ABIERTOS**: todo tipo de computadora, sistemas operativos, protocolos de red y otros, hardware y software, pueden interconectarse y trabajar coordinadamente para lograr los objetivos del usuario
- ✓ Los sistemas cliente-servidor basados en estándares abiertos apoyan la interoperabilidad.
 - Interoperabilidad se refiere a la habilidad de dos o más sistemas para intercambiar y usar software y datos.
 - Los estándares abiertos promueven un mercado de proveedores, lo que conduce a costos más bajos y mayor calidad, siendo INTERNET el área con mayor estandarización

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

- **Sistema Cliente/Servidor:** uno o más clientes y uno o más servidores, conjuntamente con un sistema operativo y un sistema de comunicación entre procesos, forma un sistema compuesto que permite cómputo distribuido, análisis y presentación de los datos.
- Si existen múltiples servidores de procesamiento de bases de datos, cada uno de ellos deberá procesar una base de datos distinta, para que el sistema sea considerado un sistema *Cliente/Servidor*.
- Los clientes a través de la red, pueden realizar consultas al servidor, y el servidor tiene el control sobre los datos, pudiendo los clientes tener datos privados que residen en sus computadoras.
- El enfoque cliente-servidor apoya el crecimiento escalable de la capacidad de hardware y software. La escalabilidad se refiere a la habilidad para agregar y remover capacidad en unidades pequeñas:
 - ✓ La escalabilidad vertical se refiere a la habilidad para agregar capacidad en el lado del servidor.
 - ✓ La escalabilidad horizontal se refiere a la habilidad para agregar capacidad en el lado del cliente a través de estaciones de trabajo adicionales y movimiento de trabajo entre clientes y servidores.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Características:

- ***Principales características de un sistema Cliente/Servidor:***

- ☐ El servidor presenta a todos los clientes una interfaz única y bien definida.
- ☐ El cliente no necesita conocer la lógica del servidor, solo su interfaz externa.
- ☐ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ☐ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

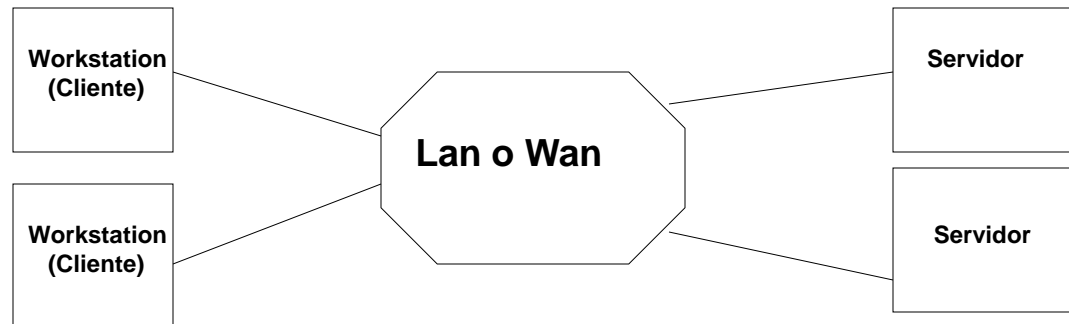
Características:

- Ejemplos de clientes:
 - ✓ Interfaces de usuarios para enviar comandos a un servidor.
 - ✓ APIs (Application Program Interface) para el desarrollo de aplicaciones distribuidas
 - ✓ Herramientas en el cliente para acceder a servidores remotos (ejemplo: servidores SQL)
 - ✓ Aplicaciones que solicitan acceso a servidores para algunos servicios
- Ejemplos de servidores:
 - ✓ Servidores de ventanas como X-windows
 - ✓ Servidores de archivos como NFS (Network File System: Sistemas de Archivos de la Red)
 - ✓ Servidores para el manejo de base de datos (servidores de SQL)
 - ✓ Servidores de diseño y manufactura asistidos por computador

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Partes de un sistema Cliente/Servidor:

- Los principales componentes de un sistema Cliente/Servidor son:
 - ✓ El núcleo (back-end o sección posterior). Es el SGBD propiamente (Servidor)
 - ✓ La Interfaz (front-end). Aplicaciones que funcionan sobre el SGBD (Cliente).



Ambiente Cliente/Servidor Generico

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Diferencia con un sistema centralizado multiusuario:

- El cliente es una terminal “con criterio”.
- El cliente tiene su propio sistema operativo y puede manejar entradas (teclado, razón, etc.) y salidas (pantalla, impresora local, sonido, etc.) sin el servidor.
- El servidor espera pasivamente la petición del cliente .
- La distribución del proceso permite al cliente ofrecer un ambiente más amigable que una terminal “sin criterio”.
- La complejidad del servidor disminuye si la comparamos con un mainframe
- El conjunto de un sistema Cliente/Servidor conduce a un ambiente flexible y dinámico.
- La parte cliente maneja la entrada de datos, acepta consulta de los usuarios y muestra los resultados.
- El servidor de la aplicación es quien procesa la consulta que le envía el cliente
- El servidor devuelve los resultados al cliente y este se los muestra al usuario.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

SECCION FRONTAL :

- **Son las diversas aplicaciones ejecutadas dentro del SGBD, tanto las escritas por los usuarios como las “integradas” que son las proporcionadas por el proveedor del SGBD o por otros proveedores de aplicaciones.**
- **No existen diferencias entre las aplicaciones escritas por los usuarios y las integradas para la sección posterior. Todas las aplicaciones utilizan la misma interfaz de usuario con la sección posterior**

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

FUNCIONES DEL CLIENTE:

- **Administrar la interfaz gráfica de usuario.**
- **Aceptar datos del usuario.**
- **Procesar la lógica de la aplicación.**
- **Generar las solicitudes para la BD.**
- **Transmitir las solicitudes de la BD al servidor.**
- **Recibir los resultados del servidor.**
- **Dar formato a los resultados.**

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

COMO TRABAJA LA SECCION FRONTAL :

- La secuencia de eventos que ocurren cuando el usuario accede al servidor de la base de datos se puede generalizar en seis (6) pasos básicos:

✓ Sistema Cliente:

- 1) El usuario crea una consulta.
- 2) La aplicación cliente da formato a la consulta y la envía al dbms.

✓ Sistema Servidor:

- 3) El servidor comprueba los derechos de seguridad del usuario
- 4) El servidor procesa la consulta y devuelve los resultados.

✓ Sistema Cliente:

- 5) El cliente recibe la respuesta y le da formato
- 6) El usuario ve y/o manipula los datos

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

COMO TRABAJA LA SECCION FRONTAL :

- El usuario crea la consulta, cualquier acción que el usuario puede hacer sobre la base de datos (pedir datos, insertar, modificar, borrar), creada en el instante, preprogramada o almacenada con anterioridad.
- La aplicación cliente convierte la consulta al SQL utilizado por el servidor de la base de datos y la envía a través de la red al servidor
- El servidor verifica que el usuario tenga los permisos apropiados para la consulta de datos que requiere, de ser así verifica la consulta y envía los datos apropiados de vuelta al cliente.
- La aplicación cliente recibe la respuesta y le da formato para presentarla al usuario
- El usuario ve la respuesta en la pantalla, puede manipular los datos y comenzar de nuevo el proceso

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de aplicaciones clientes

- Aplicaciones escritas por usuarios escritas en un lenguaje de programación convencional o un lenguaje propio (PI Sql).
- Aplicaciones o herramientas suministradas por los proveedores, para ayudar en el proceso de creación y ejecución de otras aplicaciones que realicen alguna tarea específica. La razón del uso de herramientas es para que los usuarios (finales principalmente) puedan crear aplicaciones sin escribir programas convencionales. Ej una herramienta sería un procesador de lenguaje de consultas para que los usuarios realicen las consultas a la BD.
- Clases de herramientas suministradas por los proveedores:
 - ✓ Procesadores de lenguaje de consultas
 - ✓ Generadores de reportes
 - ✓ Subsistemas de gráfica para negocios
 - ✓ Paquetes estadísticos
 - ✓ Administrar copias
 - ✓ Generadores de aplicaciones (incluyendo procesador de lenguajes de consulta de cuarta generación o 4 GL)
 - ✓ Para desarrollar aplicaciones, incluyendo productos para ingeniería de software asistida por computadora (CASE)

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

SECCION POSTERIOR :

- **Es el SGBD en si.**
- **Permite llevar a cabo todas las funciones básicas de un SGBD:**
 - ✓ **Definición de datos**
 - ✓ **Manipulación de datos**
 - ✓ **Seguridad**
 - ✓ **Integridad**
 - ✓ **Transparencia**
 - ✓ **Establecer todos los aspectos de los niveles externo, conceptual e interno (arquitectura ANSI/SPARC)**

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

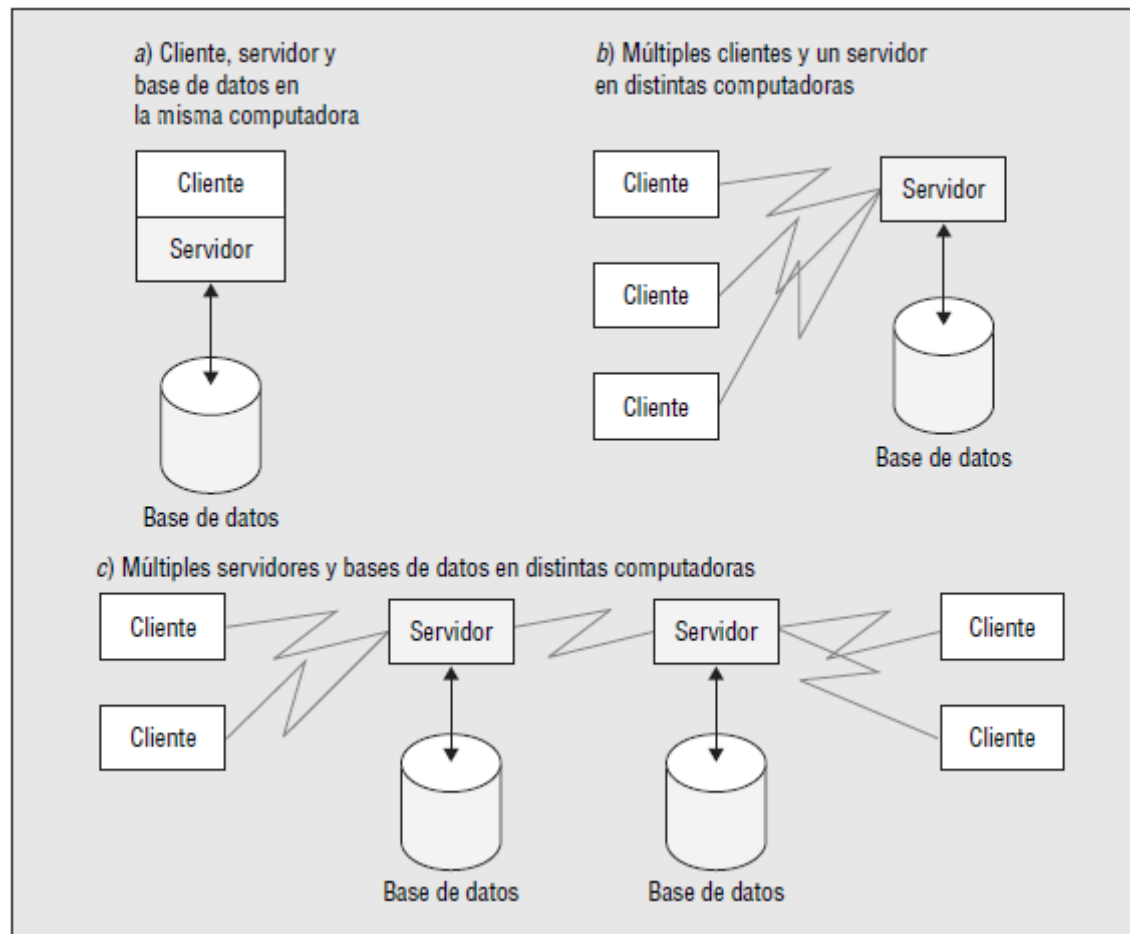
FUNCIONES DEL SERVIDOR:

- **Aceptar las solicitudes de la base de datos de los clientes**
- **Procesar las solicitudes**
- **Dar formato a los resultados y transmitirlos al cliente**
- **Llevar a cabo la verificación de integridad**
- **Mantener los datos generales de la base de datos**
- **Proporcionar control de acceso concurrente**
- **Llevar a cabo la recuperación**
- **Optimizar el procesamiento de consultas y actualizaciones**

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

- **Así, la arquitectura cliente-servidor soporta muchas formas de distribuir el software y los datos en una red de cómputo, para mejorar el desempeño y la disponibilidad de los datos**
- ✓ **Esquema simple en el cual el software como los datos en una misma computadora (a)**
- ✓ **Esquemas que aprovechan las ventajas de una red:**
 - **El software del servidor y la base de datos se ubican en una computadora remota (b)**
 - **El software del servidor y la base de datos se ubican en varias computadoras remotas.**

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR



ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

- El DBMS tiene varias responsabilidades en una arquitectura cliente-servidor; proporciona el software que se puede ejecutar en el cliente y en el servidor.
- Es común que el software del cliente sea responsable de aceptar los comandos del usuario, desplegar los resultados y realizar algún procesamiento a los datos.
- El software del servidor valida las peticiones de los clientes, localiza las bases de datos remotas, las actualiza, en caso de ser necesario, y envía los datos al cliente en un formato que éste pueda entender.
- Las arquitecturas cliente-servidor proporcionan una forma flexible para que los DBMS interactúen con las computadoras de la red. La distribución del trabajo entre clientes y servidores y las alternativas disponibles para ubicar los datos y el software tienen su complejidad

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de servidores

- ***Iterativos***
- ***Concurrentes***
- ***De Transacciones***
- ***De datos***

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de servidores iterativos

- **Pasos que realiza**
 - ✓ **Espera a que llegue la consulta de un cliente**
 - ✓ **Procesa la consulta**
 - ✓ **Envía la respuesta al cliente que envió la consulta**
 - ✓ **Vuelve al estado inicial**

Inconveniente:

Durante el tiempo en que el servidor está procesando la consulta, ningún otro cliente es servido

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de servidores concurrentes

- **Pasos que realiza**
 - ✓ **Espera a que llegue la consulta de un cliente**
 - ✓ **Cuando le llega una nueva consulta, comienza un nuevo proceso para manejar esa consulta (depende del sistema operativo la forma en como se realice este paso). El nuevo servidor maneja la totalidad de la consulta. Cuando se ha procesado completamente este nuevo proceso termina**
 - ✓ **Se vuelve al primer paso**

Ventaja:

- ✓ ***El servidor ejecuta un nuevo proceso para manejar cada consulta.***
- ✓ ***Cada cliente tiene su "propio" servidor.***
- ✓ ***Todo asumiendo que el sistema operativo permite la multiprogramación, clientes múltiples y servicio concurrente***

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de servidores de transacciones o de consultas

- Pueden proporcionar una interfaz para que los clientes envíen peticiones (acciones) al servidor
- El servidor ejecuta las peticiones y devuelve los resultados al cliente
- Los usuarios pueden especificar las peticiones con SQL
- Los usuarios pueden especificar sus peticiones mediante la interfaz de una aplicación utilizando llamadas a procedimientos remotos (RPC: “Remote Procedure Call”)
- Responde a la división funcional entre la parte visible al usuario y el sistema subyacente
- Las computadoras personales gestionan la parte visible al usuario por el procesamiento que requiere el código de la interfaz gráfica.
- Los sistemas servidores tienen almacenado un gran volumen de datos y soportan la funcionalidad del sistema subyacente

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Interacción entre clientes y servidores

- Se han desarrollado diferentes normas para la interacción entre clientes y servidores, por ejemplo ODBC (Open Database Connectivity: Conectividad abierta de base de datos), para permitir que los clientes generen instrucciones SQL para enviarlas al servidor en donde se ejecuten, así cualquier cliente que utilice interfaz ODBC puede conectarse a cualquier servidor que proporcione esa interfaz, anteriormente se necesitaba que la parte visible al usuario y al sistema subyacente fuera provista por el mismo fabricante.
-
- ODBC es un conjunto de llamadas de bajo nivel que permite a las aplicaciones clientes intercambiar instrucciones con las aplicaciones del servidor y compartir datos, sin necesidad de conocer nada unas respecto de las otras.
-
- Las aplicaciones emplean módulos llamados *controladores de bases de datos* que unen la aplicación con el SGBD elegido, empleándose SQL como lenguaje de acceso a los datos. El SGBD debe proporcionar los controladores adecuados para que puedan ser utilizados por los lenguajes de programación que soporten ODBC.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de servidores de datos

- Se utilizan en redes de area local donde se alcanza una alta velocidad de conexión entre los clientes y el servidor
- Las máquinas clientes son comparables al servidor en cuanto al poder de procesamiento y se ejecutan tareas de cómputo intensivo.
- En este entorno tiene sentido enviar los datos a las máquinas clientes, allí realizar todo el procesamiento y luego enviar los datos de vuelta al servidor
- Los clientes deben poseer todas las funcionalidades del sistema subyacente
- La arquitectura de estos servidores se ha hecho muy ‘popular’ en los sistemas de base de datos orientadas a objetos, donde se dá cabida a tipos de datos binarios para poder guardar código binario que es el que forman los objetos de sonido, imágenes, videos, etc. Los SGBDROO (SGBD Orientada a Objetos) proporcionan toda la potencia y robustez de los SGBD relacionales y permiten gestionar objetos de un modo nativo junto con los campos tradicionales (numéricos y caracteres).
- ***El costo en tiempo de comunicación entre el cliente y el servidor es alto comparado al de acceso a una memoria local (milisegundos frente a menos de 100 nanosegundos)***

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Servidores de datos: Envío de Páginas

- La unidad de comunicación de datos puede ser de una página o de una tupla.
- Si la unidad de comunicación de datos es una tupla la sobrecarga por la transferencia de mensajes es alta comparada con el número de datos transmitido.
- En lugar de enviar una sola tupla es conveniente el envío de otras tuplas que es probable se utilicen en el futuro próximo.
- Preextracción: es la acción de buscar y enviar tuplas antes de que sea estrictamente necesario.
- Al residir varias tuplas en una página, el envío de páginas puede considerarse una forma de preextracción, así cuando un proceso desee acceder a una única tupla de la página, se enviarán todas las tuplas de esta página

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Servidores de datos: Bloqueos

- El bloqueo de las tuplas de datos que el servidor envía a los clientes la realiza habitualmente el propio servidor.
- Un inconveniente del envío de páginas es que los clientes pueden recibir implícitamente el bloqueo de todas las tuplas que residen en la página (los clientes adquieren bloqueos sobre todas las tuplas preextraídas, aunque no estén accediendo a alguna de ellas)
- El bloqueo de todas las tuplas que residen en la página puede detener innecesariamente el procesamiento de otros clientes que necesitan bloquear esos elementos.
- Se han desarrollado algunas técnicas para la liberación de bloqueos, en las que el servidor pide a los clientes que le devuelvan el control sobre los bloqueos de las tuplas extraídas. Si el cliente no necesita la tupla extraída, puede devolver los bloqueos sobre esa tupla al servidor para que éstos puedan ser asignados a otros clientes

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Servidores de datos: Cache de datos

- Los datos que se envían al cliente a favor de una transacción se puede alojar en un caché del cliente, incluso una vez completada la transacción, si hay espacio de almacenamiento libre suficiente
- Las transacciones sucesivas en el mismo cliente pueden hacer uso de los datos en caché
- ***Coherencia de la caché:*** la transacción que encuentra los datos en la caché debe asegurarse que estén actualizados dado que esos datos pudieron haber sido modificados por otros clientes, entonces se debe establecer una comunicación con el servidor para comprobar la validez de los datos y adquirir un bloqueo sobre ellos.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Servidores de datos: Cache de bloqueos

- Los bloqueos, también, se pueden almacenar en la memoria caché del cliente si la utilización de los datos está prácticamente dividida entre los clientes.
- Ejemplo:
 - ✓ En la memoria caché se encuentra el elemento de dato buscado y el bloqueo requerido para acceder al mismo, así el cliente puede acceder al elemento de dato sin comunicar nada al servidor.
 - ✓ El servidor debe seguir el “rastros” de los bloqueos en caché
 - ✓ Si un cliente solicita un bloqueo al servidor, este debe comunicar a todos los clientes sobre el bloqueo sobre el elemento de datos que se encuentra en caché de otros clientes.
 - ✓ La tarea se vuelve muy complicada al tener en cuenta las posibles fallas en las máquinas
- Esta técnica se diferencia de la liberación de bloqueos en la caché de bloqueo que se realiza a través de las transacciones, de otra forma las dos técnicas serían similares

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

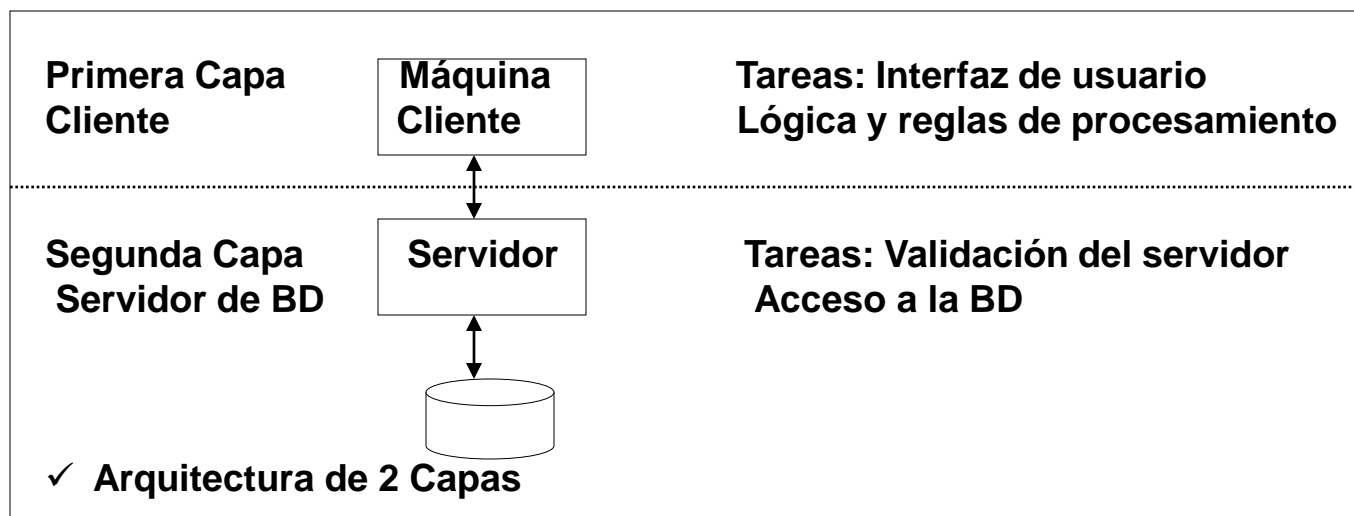
Tipos de Arquitecturas:

- ***Arquitectura de 2 capas***
- ***Arquitectura de 3 capas***

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de Arquitecturas: Arquitectura de 2 capas

- Esta es la arquitectura tradicional.
- Tiene 3 componentes:
 - ✓ Interfaz de usuario
 - ✓ Gestión del procesamiento
 - ✓ Gestión de la base de datos



ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de Arquitecturas: Hay 2 tipos de arquitectura Cliente/Servidor de 2 capas

- ***Thick Client:*** la mayor parte de la lógica de la aplicación (gestión del procesamiento) reside junto a la lógica de la presentación (interfaz de usuario) en el cliente, con la porción de acceso a datos en el servidor
- ***Thin Clients:*** solo la lógica de la presentación reside en el cliente, con el acceso a los datos y la mayoría de la lógica de la aplicación en el servidor

Es posible que un servidor funcione como cliente de otro servidor: Diseño de dos capas encadenados

Limitaciones:

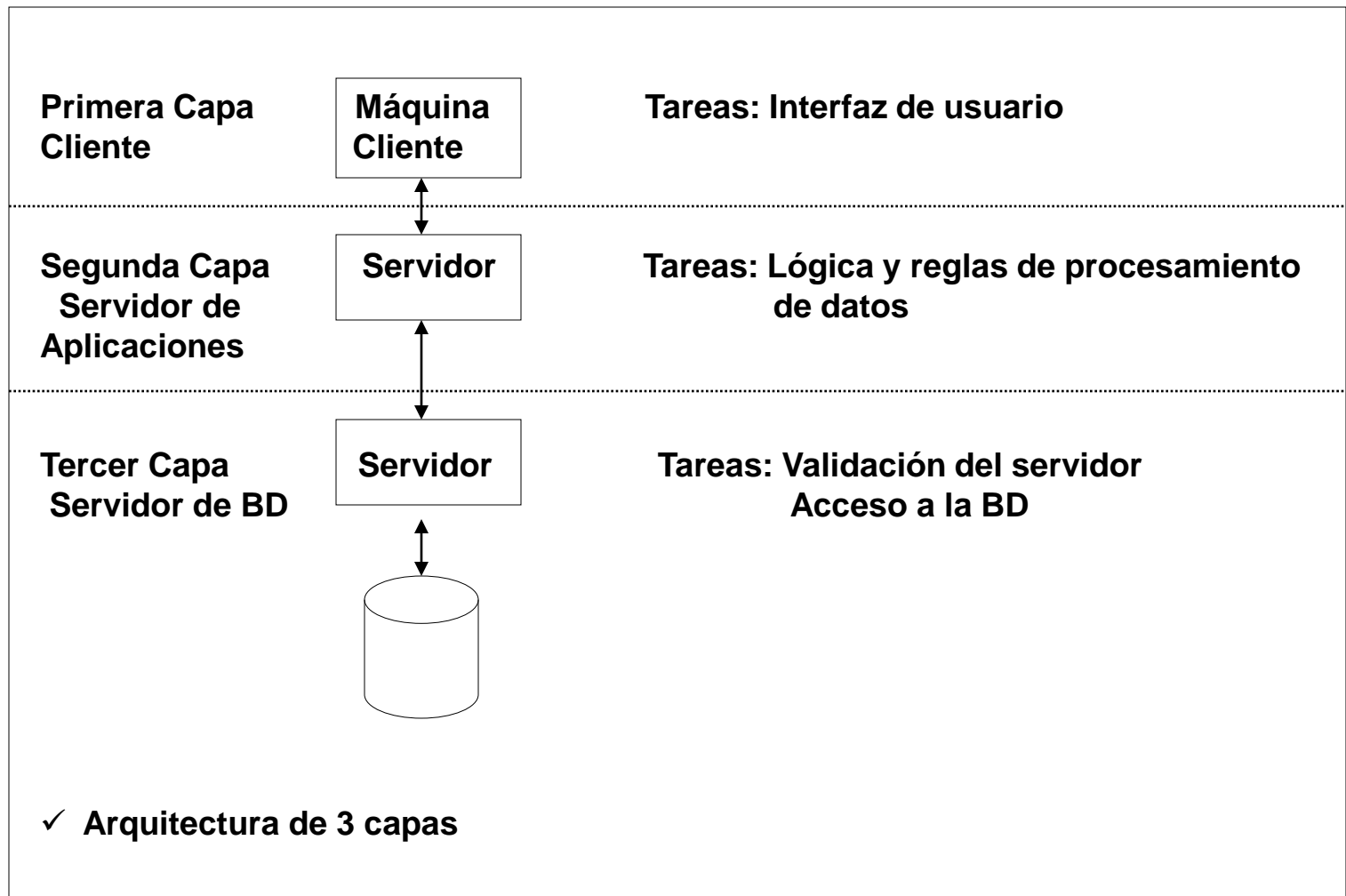
- El máximo número de usuarios es limitado, sino se excede la capacidad de procesamiento
- No hay independencia entre la interfaz de usuario y los tratamientos, lo que va en contra de la evolución de las aplicaciones
- Es dificultoso relocalizar las capas de tratamiento consumidoras de cálculo
- Bajo esta arquitectura se complica reutilizar los programas desarrollados

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de Arquitecturas: Arquitectura de 3 capas

- Surge para superar las limitaciones de la arquitectura de 2 capas
- La tercer capa Servidor intermedio está entre la interfaz de usuario (Cliente) y el gestor de la base de datos (Servidor).
- El Servidor intermedio proporciona gestión del procesamiento y en ella se ejecutan las reglas y lógica del procesamiento, permitiendo que un número “ilimitado” de usuarios interactúe en esta arquitectura
- Se utiliza para incrementar el rendimiento, la flexibilidad, mantenibilidad, reusabilidad y escalabilidad del sistema Cliente/Servidor, “escondiendo” la complejidad del procesamiento distribuido al usuario.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR



ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de Arquitecturas: Arquitectura de 3 capas

Limitaciones:

- Las herramientas de programación que soportan el diseño de arquitectura de 3 capas no proporcionan todos los servicios deseados necesarios para soportar un ambiente distribuido.
- Un problema potencial es la separación de la interfaz gráfica de usuario, la lógica de gestión de procesamiento y la lógica de datos.
- La ubicación de una función particular en una capa u otra debería basarse en criterios como los siguientes:
 - ✓ Facilidad de desarrollo y comprobación
 - ✓ Facilidad de administración
 - ✓ Escalabilidad de los servidores
 - ✓ Funcionamiento (incluyendo procesamiento y carga de la red)

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

Tipos de Arquitecturas: Middleware

- Es el software que proporciona un conjunto de servicios que permite el acceso transparente a los recursos en una red
- Es un módulo intermedio que actúa como conductor entre dos módulos de software, los cuales para compartir datos no necesitan saber como comunicarse entre ellos, sino como comunicarse con el módulo de middleware
- Es el encargado de acceso a los datos acepta las consultas y datos recuperados directamente de la aplicación y los transmite por la red
- Es el responsable de enviar de vuelta a la aplicación, los datos de interés y de la generación de los códigos de error.

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

VENTAJAS E INCONVENIENTES:

Ventajas

- ***Interoperabilidad:*** los componentes claves trabajan juntos: cliente, servidor y red
- ***Flexibilidad:*** la nueva tecnología puede incorporarse al sistema
- ***Escalabilidad:*** cualquiera de los elementos del sistema puede reemplazarse cuando es necesario, sin impactar sobre otros elementos. Si la base de datos crece, las computadoras clientes no tienen que equiparse con memoria o discos adicionales. Los cambios afectan solo a la computadora en la que se ejecuta la BD
- ***Usabilidad:*** facilita el uso a los usuarios
- ***Integridad de los datos:*** entidades, dominios e integridad referencial son mantenidas en el servidor de la base de datos
- ***Accesibilidad:*** los datos pueden ser accedidos desde múltiples clientes
- ***Rendimiento:*** se puede optimizar el rendimiento por hardware y procesos
- ***Seguridad:*** la seguridad de los datos está centralizada en el servidor

ARQUITECTURA DE BASE DE DATOS CLIENTE/SERVIDOR

VENTAJAS E INCONVENIENTES:

Inconvenientes

- Alta complejidad tecnológica al tener que integrar una gran variedad de productos
- El mantenimiento de los sistemas es complejo por la interacción de diferentes partes de hardware y de software, distribuida por distintos proveedores, dificultando el diagnóstico de fallas
- Implica el rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones, almacenamiento de datos, etc).
- No es sencillo dividir las aplicaciones entre la parte cliente y la parte servidor.
- Es más complicado asegurar un elevado grado de seguridad en una red de clientes y servidores que en un sistema con una única computadora centralizada (cuanto más distribuida es la red, mayor es su vulnerabilidad).
- Problemas de congestión en la red pueden reducir el rendimiento del sistema por debajo de lo que se obtendría con una arquitectura centralizada.
- La interfaz gráfica de usuario puede a veces enlentecer el funcionamiento de la aplicación.
- La existencia de costos ocultos pueden encarecer su implantación (nuevas tecnologías, cambios organizativos, etc)