

Arquitectura ANSI/SPARC

La arquitectura ANSI / SPARC se divide en 3 niveles denominados:

1. **EL NIVEL INTERNO** es el más cercano al almacenamiento físico. Es el que se ocupa de la forma como se almacenan físicamente los datos. (DBA).
2. **EL NIVEL EXTERNO** es el más cercano a los usuarios, es decir, es el que se ocupa de la forma como los usuarios individuales perciben los datos. REPRESENTACIONES, PUEDE HACER TANTAS VISIONES COMO USUARIOS PUEDA TENER UNA BD. (N. VISIÓN)
3. **EL NIVEL CONCEPTUAL** es un nivel de mediación entre los otros dos. (TRABAJA DEFINIENDO ESTRUCTURAS DE ALMACENAMIENTO EL DBA).

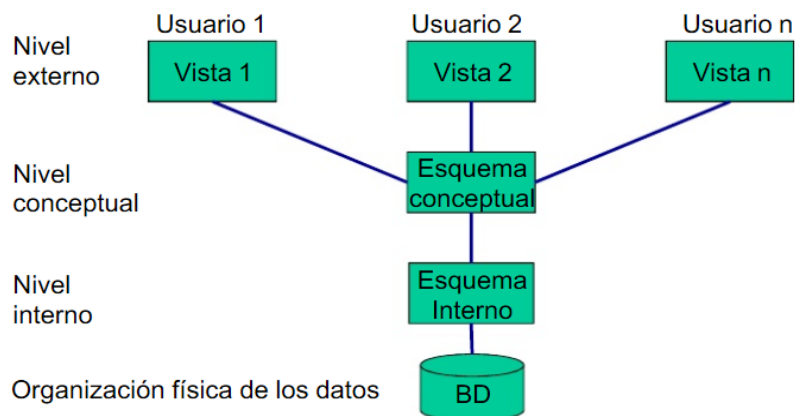
Existirán muchas “vistas externas” distintas, cada una formada por una representación más o menos abstracta de alguna parte de la BD total. Existirá sólo una “vista conceptual” formada por una representación igualmente abstracta de la BD en su totalidad.

Existirá una sola “vista interna” la cual representará a toda la BD tal como está almacenada físicamente.

EL NIVEL conceptual con toda certeza será relacional, los objetos visibles serán tablas relacionales (los operadores serán también relacionales).

EL NIVEL externo casi siempre será relacional.

EL NIVEL interno no será “relacional” porque los objetos en ese nivel no serán por lo regular sólo tablas relacionales (almacenadas), sino que serán objetos similares a los encontrados en el nivel interno de otros tipos de sistemas.



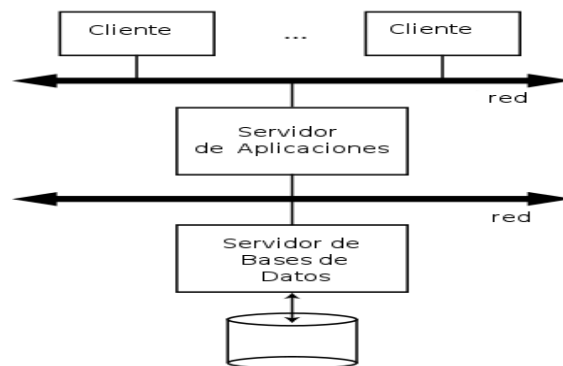
Arquitectura Cliente/Servidor

La idea principal de los sistemas Cliente/Servidor es la de diferenciar las funcionalidades que se proveen y dividir las en dos clases: funciones de servidor y funciones de cliente. Esto provee una arquitectura a dos niveles, lo que facilita el manejo de complejidad en SDBD modernos, así como la complejidad en la distribución. Esta idea surge del enfoque utilizado por los sistemas compuestos por un servidor de aplicaciones conectado a un servidor de bases de datos, donde el servidor de aplicaciones realiza algunas funcionalidades y está conectado a un sistema de bases de datos o servidor de bases de datos mediante una red de computadores. La asignación de funcionalidades entre el cliente y el servidor difiere según las implementaciones. Por ejemplo, en sistemas relacionales, el servidor realiza la mayoría de las operaciones de gestión de datos mientras que en cada nodo cliente se tiene un cliente SDBD que es puede ser responsable de la gestión de datos localmente almacenada.

Obviamente, se tienen sistemas operativos ejecutándose en ambos lados de la arquitectura y que se comunican por la red, comúnmente a nivel de sentencias SQL y respuestas o relaciones resultado.

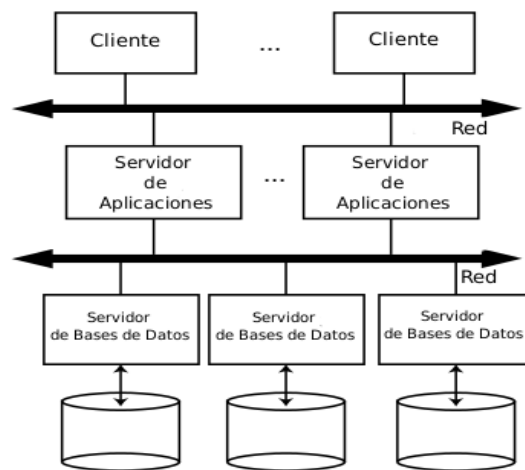
Existen distintos tipos de arquitectura modelo servidor. La implementación más sencilla es donde un servidor accedido por múltiples clientes conocido como múltiples clientes/un servidor. Este enfoque no dista de un sistema centralizado, pero presenta algunas diferencias de diseño en torno a ejecución de transacciones y manejo de cachés.

Una arquitectura más sofisticada es la de múltiples clientes/múltiples servidores, donde surgen dos alternativas para la gestión: Cualquier cliente tiene su propia conexión al servidor apropiado o cada cliente conoce cuál es su servidor local que se encarga de comunicarse con los demás servidores si fuera necesario. El primer enfoque sobrecarga el cliente de responsabilidades mientras el enfoque múltiple cliente-múltiple servidor concentra la gestión de funcionalidades a los servidores, dando como resultado que la transparencia de acceso a los datos este del lado de la interfaz de servidores y que exista el concepto de clientes ligeros.



Modelo Múltiples Cliente/Servidor

Los sistemas cliente/servidor han permitido extender distribuciones más eficientes, como por ejemplo, realizar una distribución adicional en los diferentes tipos de servidores: los clientes ejecutan la interfaz de usuario, los servidores de aplicaciones ejecutan programas de aplicación y finalmente, los servidores de bases de datos realizan las funciones de gestión de la base de datos. Bajo este enfoque, se presenta una arquitectura a tres capas, donde los nodos se organizan como servidores especializados en vez de servir como máquinas de propósito general. Adicionalmente, se tiene el diseño bajo múltiples servidores de bases de datos y múltiples servidores de aplicaciones, donde cada servidor de aplicaciones está dedicado a unas cuantas aplicaciones, mientras que los servidores de bases de datos operan en distintas formas de modo de explotar técnicas para incrementar la fiabilidad y disponibilidad, así como mejorar el desempeño de la gestión de la bases de datos en general. Esta mejora en la gestión de la base es debido a la alta integración del sistema manejador de bases de datos con el sistema operativo. Finalmente, un servidor de bases de datos puede explotar arquitecturas de hardware de última generación como los multiprocesadores o clúster de servidores para mejorar el desempeño y la disponibilidad de datos. Aunque estas ventajas son significativas, se ven afectadas por la sobrecarga de procesamiento implícito en las comunicaciones adicionales entre los servidores de aplicación y los servidores de datos. Este problema también se extiende a los sistemas cliente/servidor clásicos, pero en este caso en específico, existe una capa adicional de comunicación que no debe ignorarse.



Modelo Múltiples Clientes/Múltiples Servidores

Arquitectura Multi Bases de Datos

Los sistemas multi bases de datos representan un caso donde los SMBD individuales (distribuidos o no) son completamente autónomos y no hay concepto de cooperación, incluso pueden no conocer la existencia de otros. En nuestro caso, solo nos interesan los sistemas Multi Bases de Datos Distribuidos o Sistemas de Integración de datos (evitamos este concepto debido a que engloba sistemas que no poseen bases de datos como fuente de datos).

Las diferencias entre los Sistemas Multi Bases de Datos Distribuidos (Multi-SMBD distribuidos) y los sistemas SMBD distribuidos es evidente en sus modelos arquitecturales. Esta reside en la definición del esquema conceptual global. En los SMBD distribuidos lógicamente integrados, este esquema global define la vista global de la bases de datos entera, mientras que en el caso de los Multi-SMBD distribuidos, sólo representa una parte de las bases de datos locales que cada nodo SMBD desea compartir (esto se conoce como arquitectura de bases de datos federadas). Cada nodo debe decidir qué datos estarán disponibles, definiendo un esquema de exportación.

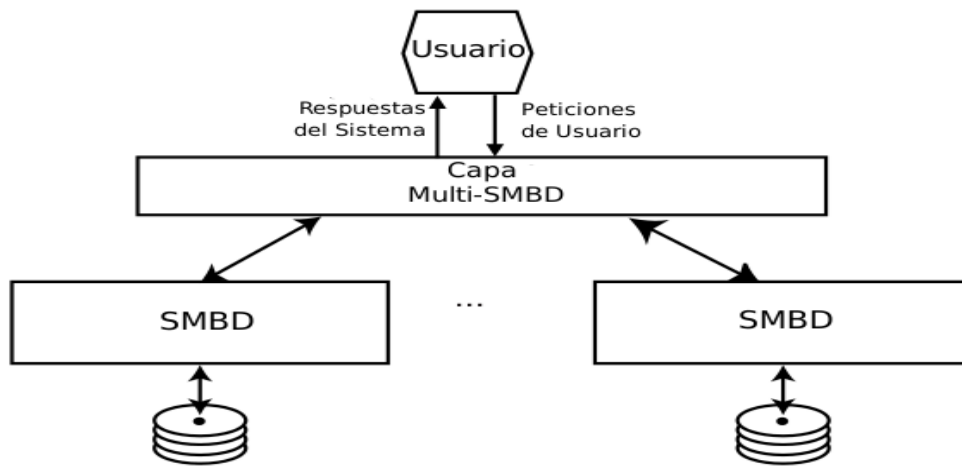
Además, la definición de una base de datos global es distinta entre un SMBD distribuido y un multi-SMBD distribuido). En los SMBD distribuidos, la base de datos global es la unión de las bases de datos locales, mientras que en los Sistemas Multi Bases de Datos Distribuidos, es sólo un subconjunto posible de la misma unión. En los Sistemas Multi Bases de Datos Distribuidos, el ECG llamado también esquema mediado, se define al integrar los esquemas externos de las bases de datos autónomas locales o posibles partes de sus esquemas conceptuales locales.

Además, los usuarios de un SMBD local definen sus propias vistas en la base de datos local y no necesitan cambiar sus aplicaciones si no desean que éstas accedan a datos desde otra base de datos. Esto obviamente genera un problema de autonomía. El diseño del Esquema Conceptual Global en los sistemas Multi Bases de Datos se refiere a la integración de los esquemas conceptuales locales (ECL) o los esquemas externos locales (EEL). Una diferencia entre el diseño en los sistemas multi bases de datos y en los SMBD distribuidos lógicamente integrados es que en los últimos el mapeo se realiza desde un esquema conceptual local al esquema conceptual global. En los sistemas multibases de datos, el mapeo se realiza en dirección contraria.

La arquitectura basada en componentes de un Sistema Multi Bases de Datos es distinta a un SMBDD. La principal diferencia es la existencia de manejadores que gestionan distintas bases de datos. El multi SMBD provee una capa que se ejecuta por encima de cada SMBD individual y provee funcionalidades al usuario para el acceso a varias bases de datos. En un MDDBS distribuido, la capa

de multi SMBD puede ejecutarse en un sólo nodo de manera centralizada o en distintos nodos para ofrecer estas funcionalidades. Finalmente, cada SMBD ignora esta capa viéndola como una aplicación que envía peticiones y recibe respuestas.

Una implementación popular de los Sistemas Multi Bases de Datos es el enfoque mediador-wrapper. Un mediador es el módulo de software que explota el conocimiento codificado acerca de ciertos conjuntos o subconjuntos de dato para crear información para aplicaciones de más alto nivel y cada mediador realiza una función en específico con interfaces claramente definidas. Para lidiar con posibles problemas de heterogeneidad, se implementan wrappers, cuya tarea es la de proveer un mapeo entre un SMBD que sirve como fuente de datos y los mediadores.



Componentes de un sistema multi SMBD