

# INSTITUTO POLITECNICO NACIONAL

Unidad Profesional Interdisciplinaria En Ingeniería Y  
Tecnologías Avanzadas



## REPORTE DE LECTURA

Grupo: 3TM3

Base de Datos distribuidas

Lic. Carlos de la Cruz Sosa

Equipo 1:

<b>Roberto Córdova Galván</b>
<b>Edgar Torres Sánchez</b>
<b>Luis Rodrigo Pedroza Velarde</b>

miércoles, 9 de marzo del 2022

## Contenido

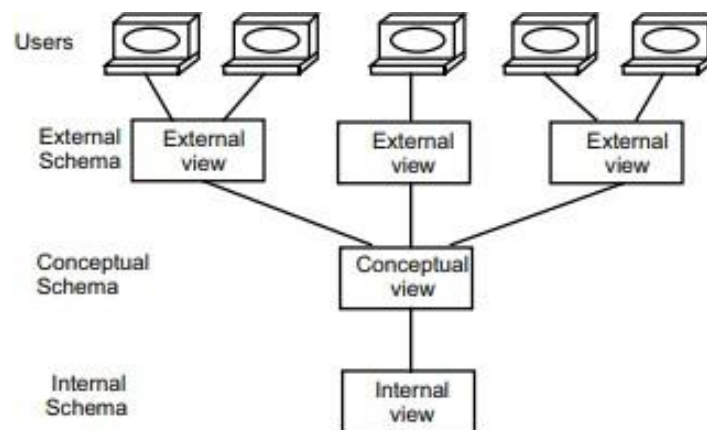
REPORTE DE LECTURA.....	1
Arquitectura ANSI/SPARC. ....	3
Una Arquitectura DBMS Centralizada Genérica. ....	0
Modelos de Arquitectura para DBMS Distribuidos.....	1
Autonomía .....	1
Distribución .....	2
Heterogeneidad .....	2
ARQUITECTURAS ALTERNATIVAS.....	2
SISTEMA DE ARQUITECTURA CLIENTE-SERVIDOR.....	2
SISTEMA DE ARQUITECTURA P2P (PEER TO PEER) .....	3
SISTEMA DE ARQUITECTURA MULTIDATABASE.....	4
Conclusiones .....	6
Bibliografía .....	7

## Arquitectura ANSI/SPARC.

La arquitectura ANSI/SPARC es un enfoque de lógica de datos para definir una arquitectura DBMS. Esta Arquitectura se enfoca en las diferentes clases y roles de usuarios y sus diferentes puntos de vista sobre los datos.

El objetivo de la arquitectura es separar los programas de aplicación de la base de datos física.

A continuación, se muestra una vista resumida del funcionamiento de esta arquitectura.



Hay tres vistas de los datos, para las cuales se va a requerir una definición de esquema adecuada (Esquema Interno, conceptual y externo) y cada una de ellas se asigna a un tipo de vista.

En el nivel más bajo de la arquitectura se encuentra la vista interna, describe la estructura física de la base de datos mediante un esquema interno, en general se ocupa de la definición física y la organización de los datos. La ubicación de los datos en diferentes dispositivos de almacenamiento y los mecanismos de acceso para la manipulación de los datos se encuentran en este nivel.

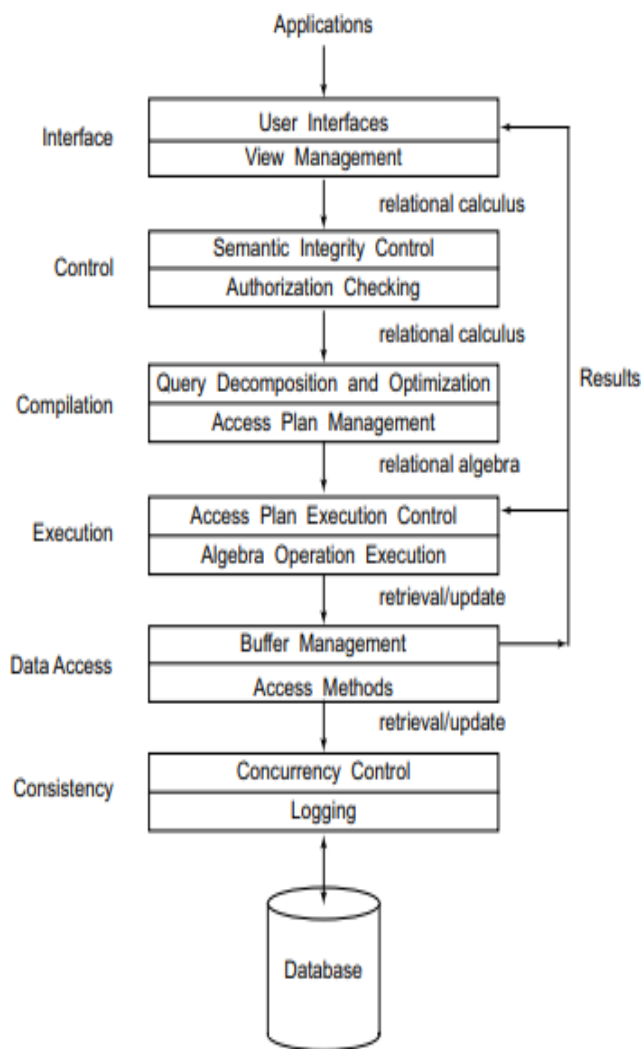
En el siguiente nivel se encuentra la vista conceptual, la de la empresa, el DBA se encuentra en este nivel y solo él lo define. Es un nivel de mediación entre el nivel interno y el externo, trabaja definiendo estructuras de almacenamiento. Es una forma de describir los datos que se almacenan en la base de datos y como se relacionan estos entre sí, pero no describe como se almacenan los datos de manera física. Se puede usar un modelo conceptual o un modelo lógico para especificar el esquema.

Finalmente encontramos la vista externa, cada esquema externo representa la parte de la base de datos a la que va a acceder el usuario, así como las relaciones que quiera ver el usuario entre los datos, sin tomar en cuenta los requisitos de las aplicaciones individuales o las restricciones de los medios físicos de almacenamiento.

## Una Arquitectura DBMS Centralizada Genérica.

Un DBMS es un programa reentrante compartido por múltiples procesos que ejecutan programas de bases de datos, al ejecutarlo en una computadora de propósito general es interconectado con otros componentes: el subsistema de comunicación y nuestro sistema operativo.

El subsistema de comunicación permite interconectar el DBMS con otros subsistemas para establecer conexión con las aplicaciones, mientras que el sistema operativo proporciona una interfaz entre el DBMS y los recursos de la computadora. A continuación, se muestran en la figura siguiente:



- Las flechas indican la dirección de los datos y el flujo de control.

- De arriba hacia abajo, las capas son la interfaz, el control, la compilación, la ejecución, el acceso a los datos y la gestión de la coherencia.

- La capa de interfaz gestiona la interfaz de las aplicaciones.

- Los programas de aplicación de bases de datos se ejecutan en vistas externas.

- La gestión de vistas consiste en traducir la consulta agregando integridad semántica y autorización. De esto se encarga la capa de control, el resultado de esta capa es una consulta en lenguaje de alto nivel.

- La capa de procesamiento de consultas asigna a la consulta una secuencia de operaciones de nivel inferior. Esta capa se ocupa del rendimiento.

- La capa de ejecución dirige la operación de los datos de acceso, incluida la gestión de transacciones y sincronización de operaciones algebraicas.
- La capa de acceso a datos gestiona las estructuras de datos que implementan los archivos, índices, etc. Gestiona los buffers de este

- modo guarda en cache los datos a los que se accede regularmente.
- La capa de consistencia administra el control de concurrencia y el registro de solicitudes de actualización.

## Modelos de Arquitectura para DBMS Distribuidos.

Usaremos clasificaciones que organizan los sistemas, tomando en cuenta la autonomía de los sistemas locales, la distribución y su heterogeneidad.

### Autonomía

En este concepto a lo que se refiere a las bases de datos distribuidas indica que cada instancia de base de datos puede operar de diferente manera e independientemente en función de los factores tales como los componentes del sistema, y además pueden ejecutar transacciones independientes y están permitidos modificarlos.

La conducta en la cual la individualidad de las bases de datos procesa queries y los optimizan no son afectados durante la ejecución global de todos y estos mismos nos permiten tener acceso a diferentes bases de datos y ya sea que la consistencia de los sistemas no debería estar comprometida cuando se le agregan más bases de datos o dejan de estar en el sistema distribuido.

Por otro lado, las dimensiones de la autonomía pueden ser especificadas como las siguientes:

- Designando la autonomía individual en las bases de datos ya que son libres de usar cualquier modelo de datos y técnicas de transacción o las que ellos prefieran
- La autonomía comunicativa en cada base de datos individual es libre de tomar cualquier decisión en qué tipo de información desea proveer a las otras bases de datos o en cualquier otro software que controla la ejecución global.
- La ejecución autónoma en cada base de datos puede ejecutar las transacciones que sean presentadas y como quieren que sean presentadas

También podemos clasificar las diferentes alternativas por importancia, una de ellas es *tight integration* donde una simple imagen de la base de datos está disponible para cualquier usuario para poder compartir información en la cual reside en una sola base de datos

La siguiente clasificación es semiautomática en la cual consiste en que las bases de datos pueden operar independientemente, pero tienen que consultar a una federación para decidir qué tipo de información puede ser compartida y además estas deciden qué base de datos puede tener acceso a cualquier usuario.

Y la última alternativa es total aislamiento donde el sistema individual está en stand-alone de las bases de datos en la que no conoce la existencia de las bases de datos adjuntas.

## Distribución

En este punto veremos cómo se puede distribuir la información, aquí tomaremos en cuenta dos modelos diferentes: cliente-servidor y peer to peer. En el modelo cliente-servidor.

En el modelo cliente-servidor concentra las actividades en el servidor mientras que el cliente está enfocado en darle un ambiente de manejo incluyendo una interfaz de usuario para que sea más fácil manipular los datos. La importancia de este modelo es que tenemos que definir quién es el cliente y quien es el servidor para poder distribuir bien los datos de las bases de datos.

El otro modelo para considerar es el peer to peer, aquí no hay distinción entre los servidores y los clientes, todas las máquinas están llenas de bases de datos que se comunican entre sí con otras máquinas para poder ejecutar queries y transacciones y casi la gran mayoría de estos sistemas manejan este tipo de distribución

## Heterogeneidad

La importancia de este punto es que el lenguaje de las queries sea solo uno y que las transacciones tengan sus protocolos de seguridad y representar la heterogeneidad con una herramienta única para el poder de un solo modelo y las limitaciones de las mismas herramientas de estas. La heterogeneidad no involucra meterse en diferentes paradigmas en los diferentes modelos de datos, pero sí cubre los diferentes lenguajes que puede manejar el sistema y que usen el mismo modelo de datos. En SQL está estandarizado para que su uso sea más fácil de manejar y además se maneje un solo modelo de datos.

## ARQUITECTURAS ALTERNATIVAS

Existen aproximadamente 18 arquitecturas de modelos de arquitectura para bases de datos distribuidas de las cuales, es importante el estudio de A0, D2, H0 para modelo cliente / servidor distribuido, arquitecturas A0, D2, H0 para modelos P2P de DBMS distribuido y A2, D2, H2 para modelos distribuidos.

## SISTEMA DE ARQUITECTURA CLIENTE-SERVIDOR

El modelo de cliente servidor nace en los años 1990's en la cual se divide la arquitectura en dos niveles la del cliente y la del servidor. Se hace una distinción del

modelo cliente servidor computacional al de manejo de base de datos, y es que, en el contexto de bases de datos, solo nos interesa saber el software que el cliente ejecuta y cual ejecuta el servidor para asistir. Obviamente también saber en cual maquina esta.

Dentro de un sistema relacional las solicitudes de consultas son procesadas sin la intención de optimizar o de entenderás, solo se envía la información al cliente que lo solicita. El servidor hace todo el trabajo y envía la información.

La forma de implementar la cantidad de servidores dentro de este modelo de arquitectura si hace la diferencia, ya que un solo servidor que recibe múltiples clientes es considerado como centralizado, pero múltiples servidores que atienden múltiples clientes es un modelo distribuido, mientras el cliente solo decide en conectarse al “servidor casa” , en donde no necesariamente todos los servidores necesitan tener sistema operativo con interfaz gráfica, pueden coexistir algunos que únicamente posean la base de datos para un mejor aprovechamiento de los recursos.

En este modelo de cliente / servidor distribuido también contempla separar o asignar a cada servidor un rol por así decirlo, por ejemplo, uno se encarga de ser el servidor de aplicación, otro de ser el servidor de base de datos, otro de ser el servidor web, de tal forma que el cliente no sabe que sus solicitudes están descentralizadas, y así evitar que una sola maquina deba realizar la ejecución de distintos servicios.

Tiene muchas ventajas el aprovechamiento del modelo del sistema cliente servidor, de los cuales es distribuir la carga de trabajo, mejorar la accesibilidad gracias al paralelismo, y explotar distintos tipos de hardware.

## SISTEMA DE ARQUITECTURA P2P (PEER TO PEER)

Otro modelo de sistema es el Peer to peer, o de extremo a extremo. Este modelo ha cambiado mucho al paso de los años y ha retomado popularidad recientemente. Los principales motivos de su uso son la distribución de miles de archivos, la heterogenia en cada punto del sistema y autonomía. Lo importante es resaltar que en cada parte del sistema se tiene la misma funcionalidad.

En este modelo de sistema P2P se utilizan tres formas de visualizar los datos, el primero es el esquema interno local el cual podemos verlo como los datos que se encuentran dentro de toda una organización en un solo edificio, por ejemplo, pero que están conectados a más edificios. Esa conexión a más edificios con distintos datos e información regida por cada organización es un esquema global de los datos. Y por último existe el tercer esquema de vista externa provee transparencia tanto de datos como a nivel de red.

Tenemos dos componentes de distribución DBMS, la primera es la de interacción con el usuario y la segunda es la de almacenamiento. En la primera consiste en cuatro puntos importantes, el primer punto para la interacción con el usuario es la interpretación de los comandos de entrada, la segunda es el control de semántica es decir que la integridad de los datos es parte de una jerarquía de autorizaciones, definidos por los tres esquemas de vista local, global y externo. La tercera actividad es la optimización de las consultas para reducir los costos de funcionamiento. Y la cuarta es distribuir la ejecución de la solicitud del usuario.

Por otra parte, del almacenamiento o también llamado procesador de datos, este componente consiste en tres acciones importantes. La primera es optimización de solicitud local, es decir acceso a la dirección buscada y elegir la mejor dirección de acceso de cualquier dato. El segundo punto es la recuperación, es decir mantener la seguridad local o integridad incluso cuando ocurra un fallo físico de hardware. Y la tercera actividad que realiza es el procesar en tiempo real accesos a la base de datos, este soporte de acceso contiene un buffer el cual es responsable de mantener la memoria en buen funcionamiento.

## SISTEMA DE ARQUITECTURA MULTIDATABASE

En este modelo cada nodo es completamente autónomo al punto de ni siquiera saber que existen los demás nodos (llámese nodo a los servidores, las máquinas que alojan los servicios), se menciona la ruta de acceso haciendo referencia a la estructura de datos utilizado y sus algoritmos para hallar los datos. El esquema global en este sistema muestra una vista de la base de datos completa el DBMSs debe elegir el permiso de vista por otros. Existe heterogeneidad en este sistema multidatabase. Diseñar el esquema conceptual global involucra agregar el esquema local y externo.

Cuando existe heterogeneidad hay dos posibilidades de implementación, la primera es unilingüe y la segunda es multilingüe.

La implementación unilingüe usa diferentes modelos de datos y lenguajes cuando se accede a la base de datos local o la global. Esto quiere decir que el tipo de usuario que accede a lo global puede ser diferente al que accede a lo local, en el sentido de utilizar distintos modelos de datos.

La alternativa de arquitectura multilingüe tiene como filosofía permitir el acceso a lo global desde el exterior. Las diferencias entre ambas son la declaración del esquema externo, se asume que es puramente local y la solicitud query es procesada como cualquier DBMSs centralizado. Se requiere un mapeo del esquema global conceptual.



La arquitectura multi-DBMS es muy diferente al DBMS distribuido. La principal diferencia es estar completamente establecido en DBMSs que cada uno contiene diferentes bases de datos.

## Conclusiones

En este trabajo concluimos en general que los sistemas de DBMS Distribuidos tienen distintas arquitecturas entre las cuales se encuentra una principal denominada ANSI/SPARC la cual es denominada como una estructura de 3 niveles las cuales se enfocan en la parte donde se encuentran los datos físicamente (nivel interno) la parte de la base que se muestra a los usuarios y ocultando datos que no son necesarios que vean (nivel externo) y por ultimo tenemos al DBA que se encargará de hacer una conexión entre los dos niveles anteriores. Cada nivel tiene su propio esquema.

Podemos decir que estos tres conceptos son muy importantes para poder diseñar sistemas distribuidos ya que tratamos de que se maneje un solo modelo ya sea enfocado en cliente-servidor y además trabajar en un solo modelo de datos para que sea más fácil su manejo y estén en heterogeneidad

Existen distintos modelos de arquitecturas que se pueden abstraer e implementar en los modelos de base de datos, y cada una posee ventajas, usos y visualizaciones completamente distintos, pero en esta lectura se enfoca en cliente-servidor, en P2P (Peer to peer) y modelo multi base de datos. Las tres poseen vistas locales, globales y externas y manejan los datos y poseen una autonomía por servidor distintas.

## Bibliografía

[1] M. T. Özsu y P. Valduriez, *Principles of Distributed Database Systems, Third Edition*. New York, NY: Springer New York, 2011. Accedido el 10 de marzo de 2022. [En línea]. Disponible: <https://doi.org/10.1007/978-1-4419-8834-8>

[2]"Modelo ANSI SPARC (ANSI PARK) Arquitectura de 3 Niveles – Bases de Datos". Fernando Arciniega. <https://fernandoarciniega.com/modelo-ansi-sparc-ansi-park-arquitectura-de-3-niveles-bases-de-datos/> (accedido el 10 de marzo de 2022).

[3] Elmasri Navathe, *Database Systems*, 7<sup>th</sup> edition. Boston, Massachusetts: Pearson, 2017.