

# Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

## **Alumno**

Gamboa Del Ángel Alan Eduardo  
Granados Bello Martin Alejandro

## **Grupo**

3TM3

## **Unidad de Aprendizaje:**

Bases De Datos Distribuidas

## **Profesor**

Carlos De La Cruz Sosa

## **Actividad / Práctica**

**Reporte Arquitecturas Bases de Datos  
Distribuidas**

## Introducción

La arquitectura de un sistema es probablemente la parte mas fundamental de este, ya que define su estructura y se identifican todos los componentes y las funciones correspondientes. La especificación de la arquitectura de cualquier sistema requiere la identificación de los módulos y las relaciones entre ellos. A continuación se presentan distintos tipos de Arquitecturas Distribuidas para las Bases de Datos.

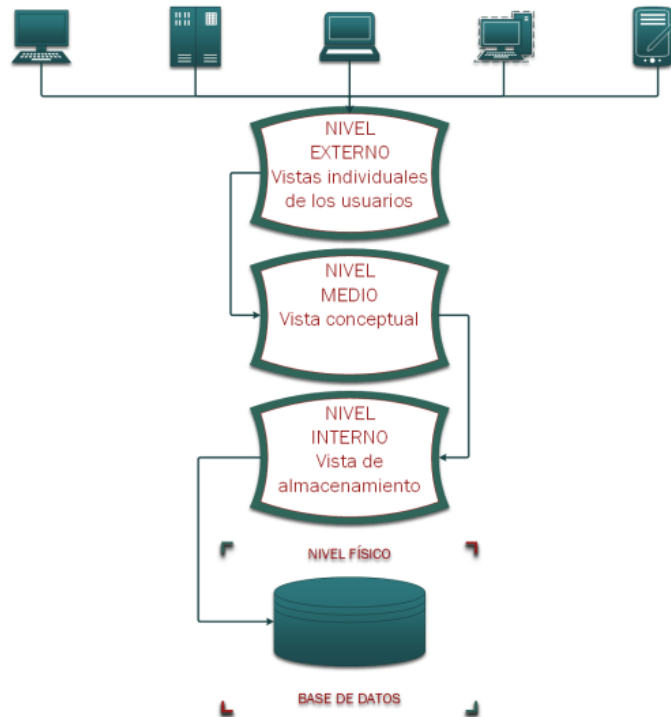
### *Arquitectura ANSI/SPARC*

La Arquitectura o Modelo ANSI-SPARC (American National Standards Institute, Standards Planning and Requirements Committee), es un estándar de diseño abstracto para un sistema de gestión de bases de datos, fue propuesta por primera vez en el año de 1975 por la empresa ANSI.

El objetivo de esta arquitectura, también llamada Arquitectura de tres niveles, es separar la vista de los usuarios, lo hace de la siguiente manera:

- **Permite vistas de usuario independientes y personalizadas:** Cada usuario debe ser capaz de acceder a los datos, pero tiene una vista personalizada diferente de los datos. Éstos deben ser independientes: los cambios en una vista no deben afectar a las demás.
- **Oculto los detalles físicos de almacenamiento a los usuarios:** Los usuarios no deberían tener que lidiar con los detalles de almacenamiento de la base de datos.
- **El administrador de la base de datos debe ser capaz de cambiar las estructuras:** Modifica la estructura de almacenamiento de la BD sin afectar la vista de los usuarios.
- **La estructura interna de la base de datos no debería verse afectada por cambios en los aspectos físicos del almacenamiento:** por ejemplo, un cambio a un nuevo disco.

Los tres niveles que maneja este modelo se reducen en el siguiente esquema:



### **Nivel externo (Vistas individuales de los usuarios)**

Una vista de usuario describe una parte de la base de datos que es relevante para un usuario en particular. Excluye datos irrelevantes, así como los datos que el usuario no está autorizado a acceder.

### **Nivel Medio (Vista conceptual)**

El nivel conceptual es una forma de describir los datos que se almacenan dentro de la base de datos y cómo los datos están relacionados entre sí. Este nivel no especifica cómo se almacenan físicamente los datos.

Algunos datos importantes acerca de este nivel son:

- El DBA (Administrador de la base de datos) trabaja en este nivel.
- Describe la estructura de todos los usuarios.
- Sólo el DBA puede definir este nivel.
- Visión global de la base de datos.
- Independiente de hardware y software.

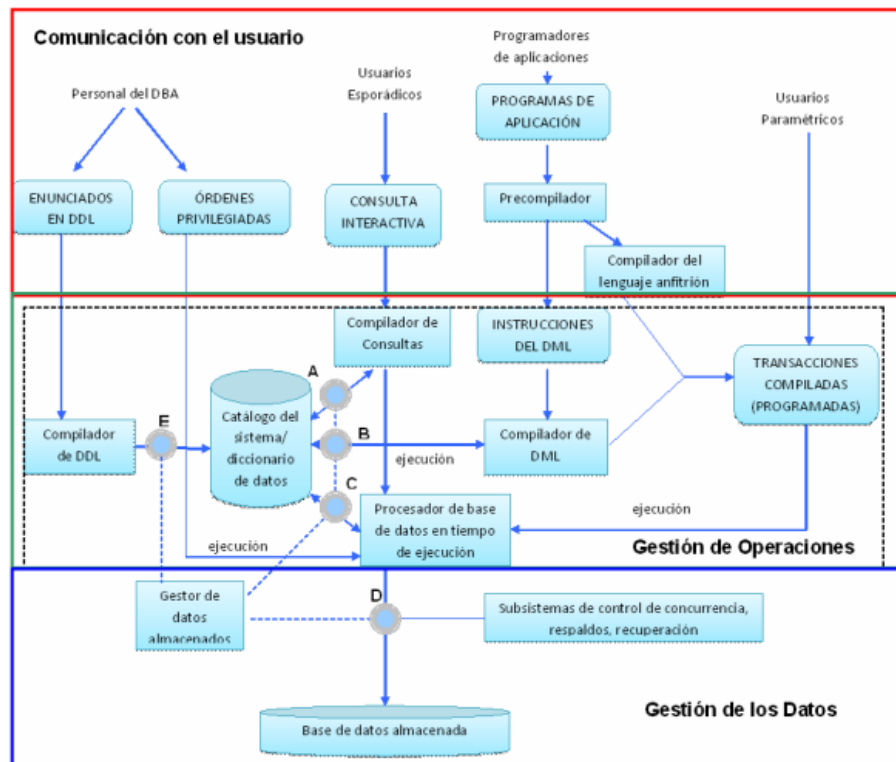
### **Nivel interno (Vista de almacenamiento)**

El nivel interno implica la forma en que la base de datos se representa físicamente en el sistema informático. En él se describe cómo los datos se almacenan en la base de datos y en el hardware del equipo.

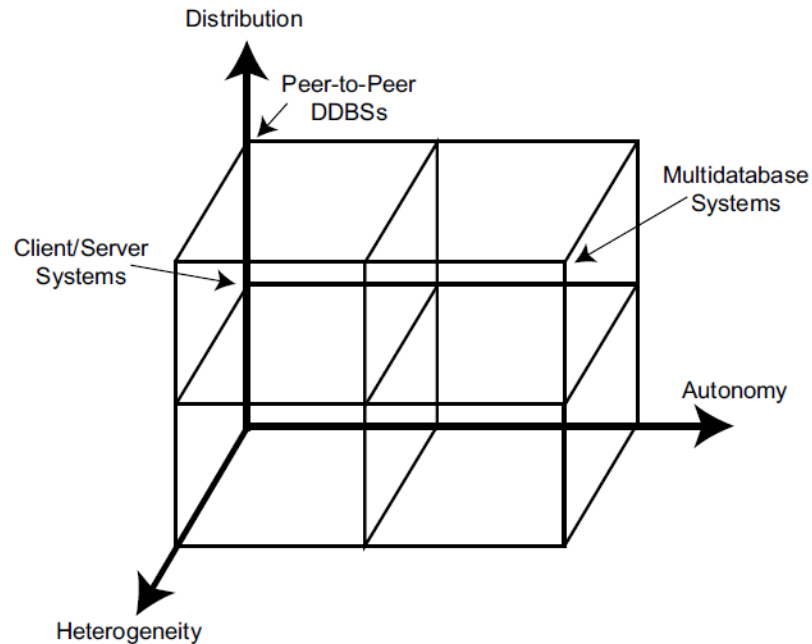
### *DBMS*

Un DBMS es un programa que comparte múltiples procesos que ejecutan programas de bases de datos. El propósito general de estos sistemas es el de manejar de manera clara, sencilla y ordenada un conjunto de datos. Estos sistemas se dividen en las siguientes capas:

- **Nivel interno, o de máquina:** Describe en detalle la forma de cómo se almacenan los datos en los dispositivos de almacenamiento. Responde a las cuestiones de rendimiento planteadas al hacer el diseño físico de la BD y al ajustarlo posteriormente a nuevas necesidades.
- **Nivel externo, o de usuario:** Es lo que el usuario final puede visualizar del sistema terminado, muestra sólo una parte de la base de datos al usuario acreditado para verla.
- **Nivel conceptual:** Describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño.



## Modelos de Arquitectura para DBMS



- **Autonomía:** Se refiere a la distribución del control. Indica el grado en el cual cada DBM puede operar de manera independiente. Es una función la cual determina si pueden ejecutar procesos por si solos y sin interferir o ser modificados por otros.

Los requerimientos para que sistema sea autónomo son los siguientes:

1. Las operaciones individuales de los DBMs no serán afectadas por su participación en el sistema distribuido.
2. La maneras en la cual cada proceso de los DBMs son consultados y optimizados no deben de ser afectados por la ejecución de consultas globales para acceder múltiples bases de datos.
3. La consistencia del sistema y de las operaciones no deben de comprometerse cuando los DBMs se unan o abandonen el sistema distribuido.

En otro sentido, las dimensiones de la autonomía se especifican a continuación:

1. **Autonomía de diseño:** Cada manejador de bases de datos son libres de usar los modelos de información y técnicas de manejo de transacciones que ellos prefieran.
2. **Autonomía de comunicación:** Cada entidad del manejador de base de datos es libre de tomar sus decisiones con respecto a que tipo de información quiere proveer a otros manejadores o al software que controla su ejecución global.
3. **Autonomía de ejecución:** Cada manejar de base de datos puede ejecutar las transacciones que son presentados de cualquier manera que desee.

## Heterogeneidad

Puede haber muchas clases de heterogeneidad tratándose de sistemas de bases de datos distribuidas, puede que haya heterogeneidad en cuanto a hardware, modelo de base de datos, lenguaje de consultas, etc.

## Arquitecturas

Existen múltiples arquitecturas posibles, no todas son relevantes, sin embargo, existen tres arquitecturas alternativas que si lo son.

- cliente/servidor.
- peer-to-peer.
- sistema de base de datos múltiple.

### Cliente/servidor

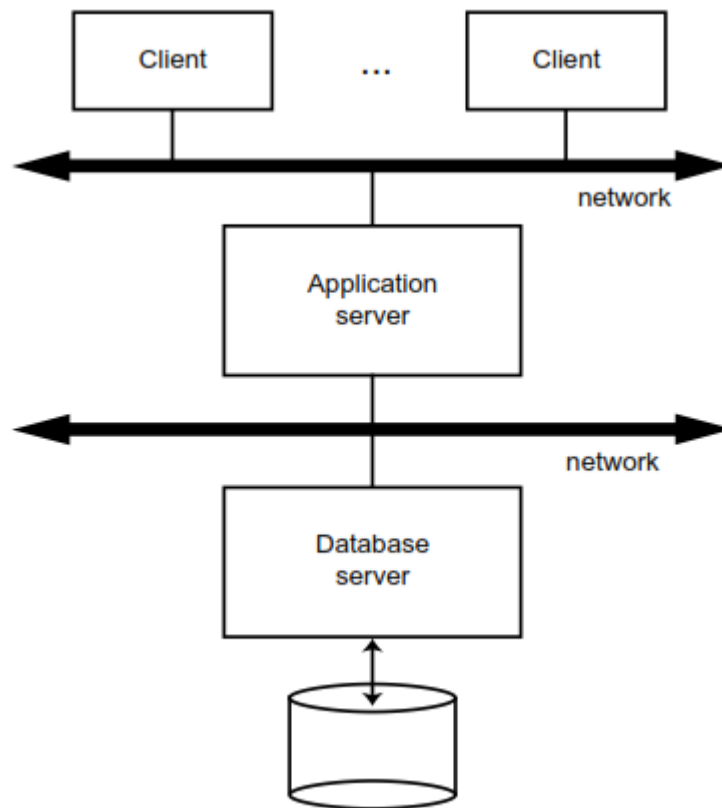
Representa un sistema heterogéneo y distribuido de múltiples bases de datos. Distinguir la funcionalidad que debe proporcionarse y dividir estas funciones en dos clases:

- Funciones de servidor
- Funciones del cliente.

Con estos dos niveles facilita la gestión de la complejidad de los DBMS modernos y la complejidad de la distribución. Si uno toma una visión centrada en el proceso, entonces cualquier proceso que solicita los servicios de otro proceso es su cliente y viceversa, pero en este caso no se tienen procesos si no máquinas reales.

En los sistemas relacionales, el servidor realiza la mayor parte del trabajo de gestión de datos. Esto significa que todo el procesamiento y la optimización de consultas, la gestión y el almacenamiento de transacciones la gestión se realiza en el servidor.

El cliente, además de la aplicación y el interfaz de usuario administrar los datos que se almacena en caché en el cliente y (a veces) administrar los bloqueos de transacciones que pueden han sido almacenados en caché también.





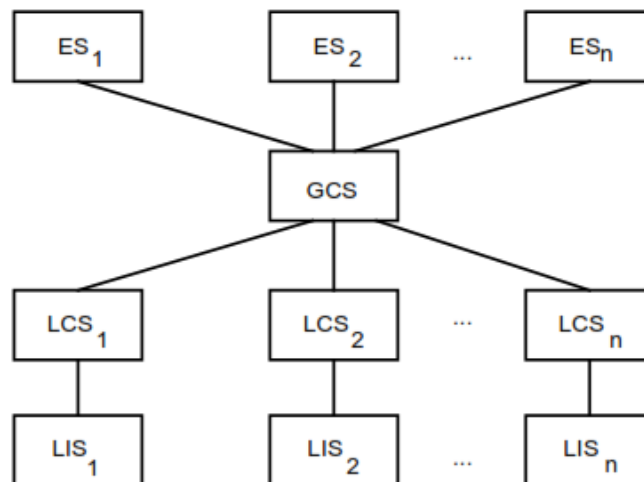
## Sistemas peer to peer

Masiva distribución en los sistemas actuales. Los principios y técnicas fundamentales de estos sistemas son muy similares a los de los sistemas cliente/servidor.

la organización física de los datos en cada máquina diferente.

Esquemas o capas en la arquitectura:

- LIS: Esquema interno local. Esquema interno individual definición en cada sitio.
- GCS: Esquema conceptual global. Describe la estructura lógica de los datos en todos los sitios.
- LCS: Esquema conceptual local. manejar la fragmentación y replicación de datos esquema conceptual global: Es la unión de los esquemas conceptuales locales.
- ES: Esquemas externos. aplicaciones de los usuarios y el acceso de los usuarios a la base de datos son compatibles con esquemas externos.



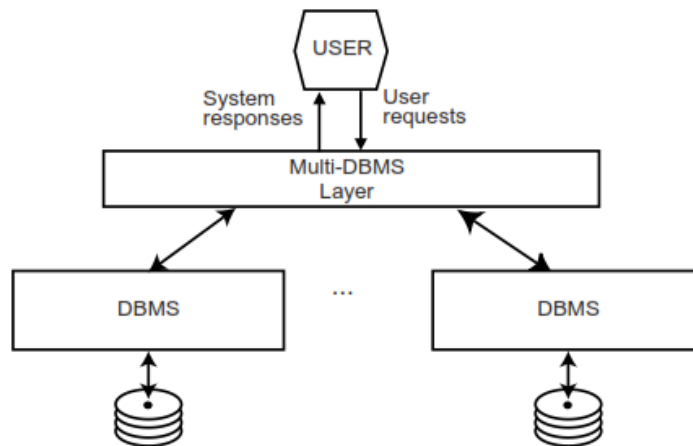
## Arquitectura del sistema de base de datos múltiple

Representan el caso en el que los DBMS individuales (ya sea distribuidos o no) son totalmente autónomos y no tienen concepto de cooperación; que puede ni siquiera “saber” de la existencia del otro o cómo hablar entre ellos.

Es diferente al termino sistemas de integración de datos ya que estos consideran datos que no son de base de datos como fuentes.

Las diferencias en el nivel de autonomía entre los multi-DBMS distribuidos y los DBMS distribuidos también se reflejan en sus modelos arquitectónicos.

En el caso de los DBMS distribuidos lógicamente integrados, el esquema conceptual global define la vista conceptual de toda la base de datos, mientras que en el caso de multi-DBMS, representa solo la colección de algunas de las bases de datos locales que cada DBMS local quiere compartir.



**Alejandro:**

Es increíble como existen diversas planificaciones, modelos, paradigmas, etc. que permiten el uso de las bases de datos distribuidas y como estas siguen avanzando aún más debido a la demanda que desde un inicio han tenido, pero principalmente por las empresas. Debido a esto es que pienso que tienen una gran ventaja competitiva frente a los sistemas centralizados, siempre y cuando la empresa tenga el conocimiento sobre cómo utilizar este tipo de base de datos y sobre todo la necesidad de usar una así.

**Alan Eduardo Gamboa Del Ángel**

Los sistemas distribuidos se han convertido en la arquitectura más utilizada en la actualidad para diseñar y construir sistemas. La arquitectura distribuida permite realizar procesos con mayor velocidad y eficiencia, disponiendo de una gran tolerancia a fallos, pues la caída de uno de los equipos que la forman no interrumpe el funcionamiento general del sistema. Sin embargo, la gestión de este tipo de sistemas es mucho más compleja que la realizada en sistemas centralizados.

## Referencias

- Android, T., 2022. ¿Qué es un DBMS? | Gestión de bases de datos - Tecnología Android. [online] Tecnología Android. Available at: <<https://tecnologiandroid.com/que-es-un-dbms-gestion-de-bases-de-datos/>> [Accessed 9 March 2022].
- M, Tamer Ozsü. Principles of Distributed Database Systems. 3rd. Ed. (Spring Street, New York, NY 10013, USA)). Pearson Education, 2011.