



GOBIERNO DEL
ESTADO DE MÉXICO



UNIVERSIDAD MEXIQUENSE DEL BICENTENARIO

**UNIDAD DE ESTUDIOS SUPERIORES
SAN JOSÉ DEL RINCÓN**

BASES DE DATOS DISTRIBUIDAS FUNDAMENTOS TEÓRICOS

CARRERA:
INGENIERÍA EN SISTEMAS COMPUTACIONALES

PRESENTA:
IMANOL CRUZ CLEMENTE

ASESOR:
I.S.C. EDUARDO BECERRIL ROMERO

Julio de 2025.

UNIVERSIDAD MEXIQUENSE DEL BICENTENARIO



GOBIERNO DEL
ESTADO DE MÉXICO



**UNIDAD DE ESTUDIOS SUPERIORES
SAN JOSÉ DEL RINCÓN**

**ANOTAR EL TÍTULO DEL TRABAJO QUE SE PRESENTA,
TAL Y COMO FUE REGISTRADO**

CARRERA:
INGENIERÍA EN SISTEMAS COMPUTACIONALES

PRESENTA:
IMANOL CRUZ CLEMENTE

ASESOR:
I.S.C. EDUARDO BECERRIL ROMERO

Julio de 2025..

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	V
PLANTEAMIENTO DEL PROBLEMA	VI
JUSTIFICACIÓN	VII
OBJETIVOS	VIII
CAPÍTULO I: MARCO TEÓRICO	IX
1. Fundamentos de base de datos distribuidas.	10
2. Diseño de una Base de Datos Distribuida	14
3. Procesamiento de consultas distribuidas	19
4. Manejo de transiciones.....	23
CAPÍTULO II: METODOLOGÍA.....	26
5. Herramientas para las bases de datos distribuidos	27
CONCLUSIONES.....	XXXIII
BIBLIOGRAFÍAS	XXXIV

Índice de figuras

Figura. 1 Representación de una base de datos distribuida	11
Figura. 2 los tres niveles de visión de usuarios	14
Figura. 3 Esquema del Modelo Cliente Servidor	28
Figura. 4 Base de datos distribuida homogénea	30

INTRODUCCIÓN

En la actualidad, el crecimiento en el almacenamiento de datos, la necesidad de acceso remoto y un mejor manejo de la información han impulsado la creación de nuevas tecnologías que permiten superar ciertas limitaciones de las bases de datos tradicionales. Así es como surgen las bases de datos distribuidas.

Estas bases de datos permiten almacenar y administrar información que está ubicada en diferentes sitios a lo largo de la red, pero que se presenta al usuario como una base de datos única.

PLANTEAMIENTO DEL PROBLEMA

JUSTIFICACIÓN

OBJETIVOS

Objetivo general:

- Comprender el funcionamiento de las bases de datos distribuidas, así como sus características, componentes y beneficios, para poder aplicarlos en el diseño y análisis de sistemas que manejan grandes cantidades de información.

Objetivos específicos:

- Identificar los conceptos fundamentales de las bases de datos distribuidas y su diferencia con las bases de datos tradicionales.

CAPÍTULO I: MARCO TEÓRICO

1. Fundamentos de base de datos distribuidas.

Chandima Jayamina. (2024, June 3) nos dice que una base de datos distribuida es una colección de múltiples bases de datos interconectadas, que están distribuidas físicamente en varias ubicaciones que se comunican a través de una red informática.

Esto nos dice que los datos de la base de datos están distribuidos en diferentes sitios a lo largo de la red informática, esto con la finalidad de almacenar una gran cantidad de datos en todos los servidores, este proceso es mejor conocido como fragmentación o partición de datos.

1.1 Conceptos básicos.

Una base de datos distribuida debe cumplir ciertos requisitos para ser determinada como tal:

- La interconexión de nodos de bases de datos debe ser a través de una red informática: los nodos deben estar conectados por una red subyacente que permita transmitir la información y los comandos entre ellos.
- Posibles diferencias entre nodos conectados: los nodos no necesariamente tienen que ser idénticos o iguales en términos de datos.
- Nodos relacionados lógicamente: es fundamental que los datos almacenados en diferentes nodos tengan una relación lógica estructurada.

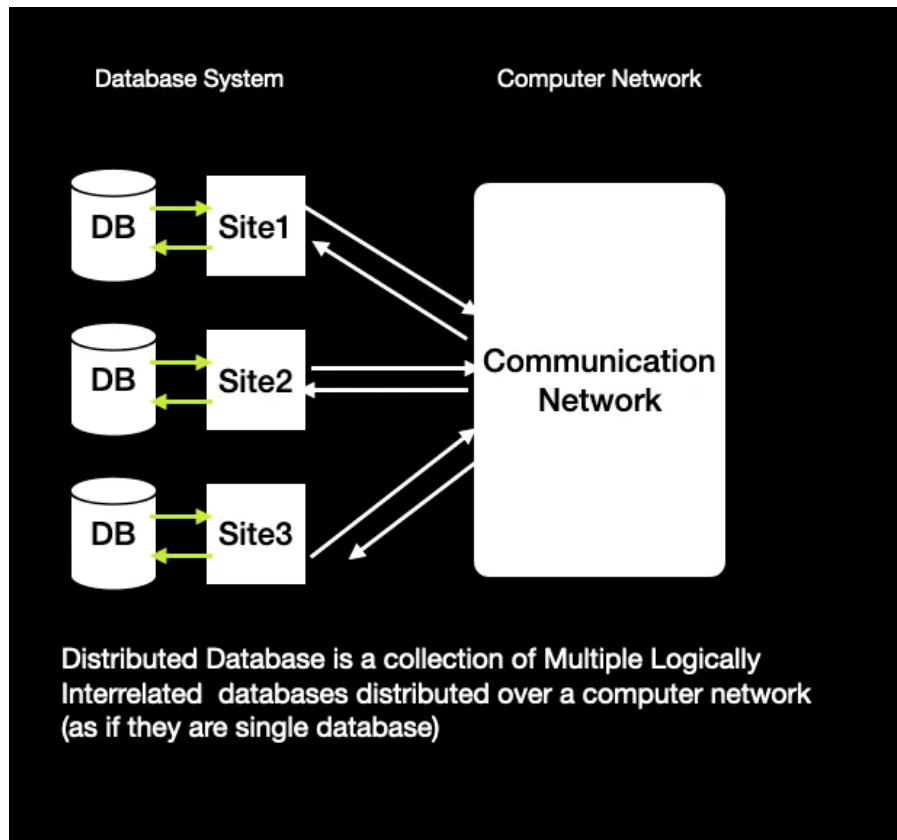


Figura. 1 Representación de una base de datos distribuida

Obtenida de: <https://chandimajayamina.medium.com/distributed-database-concepts-overview-d6c4e85fb4b5>

Todos los sitios pueden estar ubicados en proximidad física: dentro del mismo edificio o un grupo de edificios adyacentes y conectados a través de una red de área local, o pueden estar distribuidos geográficamente a grandes distancias y conectados a través de un larga distancia o red de área amplia, plantea Chandima Jayamina. (2024, June 3).

1.2 Objetivos de las bases de datos distribuidas

Toda base de datos distribuida debe estar diseñada con base en los 12 objetivos planteados por C. J. Date (Date, 2001) en 1987. A continuación se muestran los mas importantes:

1. **Independencia del nodo local:** cada nodo debe ser independiente, autónomo, y responsable de la seguridad, gestión de almacenamiento y la recuperación de información.
2. **Independencia del nodo central:** Ningún nodo es dependiente de un nodo master, o cualquier otro nodo (todos los nodos tienen los mismos privilegios).
3. **Independencia a fallas:** El sistema no se detiene por fallas en los demás nodos, el sistema sigue en funcionamiento.
4. **Transparencia de ubicación:** el usuario visualiza una sola base de datos, la fragmentación y ubicación de los datos no lo necesita saber el usuario.
5. **Transparencia de fragmentación:** el usuario solo tiene acceso a visualizar una base de datos lógica. El DDBMS selecciona, accede y maneja todos los fragmentos que se muestran al usuario.

1.3 Disciplinas de estudio

1. Sistemas Distribuidos

Un sistema distribuido es una colección de computadoras independientes que se presentan ante los usuarios como un sistema único y coherente” (Tanenbaum & Van Steen, 2007).

Estos sistemas forman la base sobre la que funcionan las bases de datos distribuidas, ya que permiten que múltiples nodos trabajen juntos.

2. Administración de Bases de Datos

Aquí se estudia cómo gestionar toda la información: desde cómo se guarda, cómo se accede, cómo se protege y cómo se organiza. En bases distribuidas esto se complica porque la información no está toda en un mismo lugar.

3. Redes de Computadoras

La infraestructura de red es esencial para la operación eficiente de bases de datos distribuidas (Kurose & Ross, 2017).

Como los datos están en distintas partes, se necesita una red rápida y segura que permita conectar todo.

4. Seguridad Informática

Se enfoca en proteger los datos de accesos no autorizados. En bases distribuidas es más complejo porque hay varios puntos de entrada y salida.

5. Sistemas de Archivos Distribuidos

Estos sistemas permiten que varios equipos accedan y gestionen archivos como si estuvieran en un solo lugar, algo muy útil para bases distribuidas.

1.4 Arquitectura de una base de datos distribuida

En una base de datos distribuida, se pueden distinguir tres niveles de arquitectura, según desde qué punto de vista se vea y qué funciones realiza el usuario:

1. Nivel interno: Aquí es donde se define cómo están guardados realmente los datos en la computadora: en qué archivos, cómo están organizados, y dónde se ubican.
2. Nivel conceptual: En este nivel se representa la estructura general de los datos, sin preocuparse por cómo se almacenan físicamente.

3. Nivel externo: Aquí se muestra solo la parte de los datos que interesa a cada usuario o aplicación.

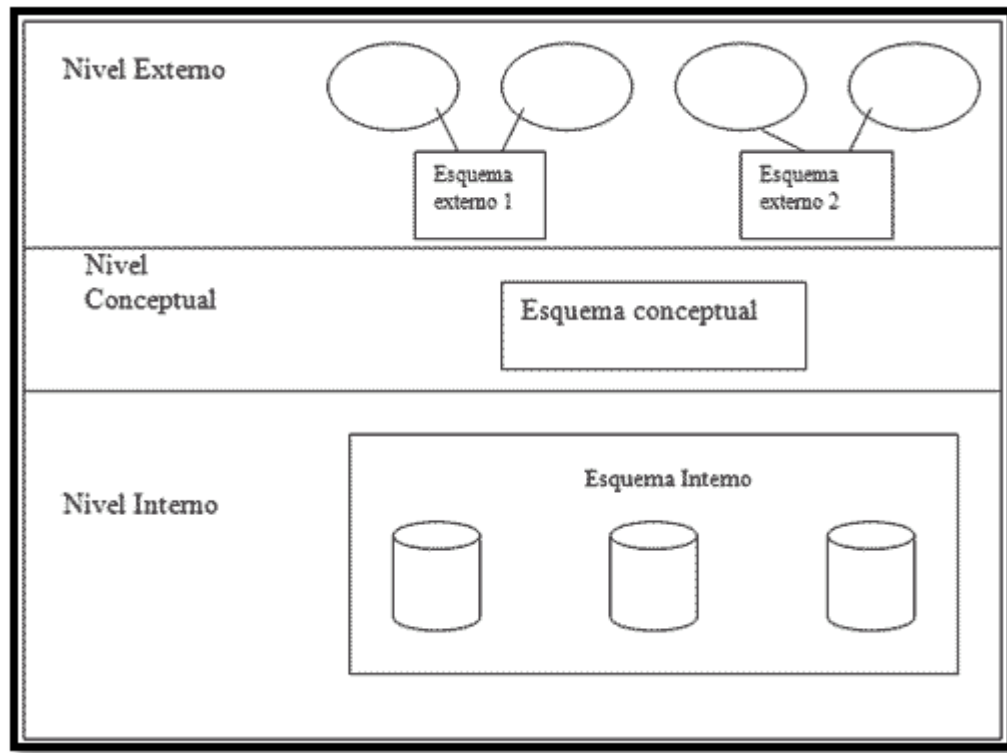


Figura. 2 los tres niveles de visión de usuarios
Obtenido de: <http://itpn.mx/recursositcs/5semestre/basededatosdistribuidas/Unidad%20I.pdf>

2. Diseño de una Base de Datos Distribuida

El problema de diseño de bases de datos distribuidos se refiere, en general, a hacer decisiones acerca de la ubicación de datos y programas a través de los diferentes sitios de una red de computadoras. Diseño de base de datos distribuidas. (2016).

Los pasos a seguir para diseñar una base de datos distribuida:

1. Diseño del esquema conceptual: hacemos un plano general de toda la información, aquí se piensa qué datos se van a guardar, cómo se van a organizar y cómo se conectan entre sí, sin preocuparse aún por cómo o dónde se van a guardar físicamente.

2. Diseño físico de la base de datos: pasamos de un plano a la ejecución, ya tenemos definido el esquema conceptual ahora toca decidir donde se guardara la información y como acceder a ella rápidamente.
3. Diseño de la fragmentación: Este paso trata de partir la información en pedazos más pequeños para trabajar mejor con los datos., se subdividen en fragmentos horizontales, verticales o mixtos.
4. Diseño de la asignación de los fragmentos: Ya que partiste la información en pedazos (fragmentos), ahora toca decidir en qué lugar físico va cada pedazo. Es como si tuvieras varias cajas con información, y ahora decides en qué estante o en qué cuarto las vas a poner.

2.1 Consideraciones de diseño de una Base de Datos Relacionales.

La metodología de diseño de bases de datos relacionales se ha consolidado a lo largo de los años satisfaciendo las propiedades de generalidad (independencia de la plataforma hardware/software), calidad del producto (precisión, completitud y eficacia) y facilidad de uso, Universidad Complutense de Madrid (2016).

Consta de las siguientes etapas:

- **Diseño conceptual:** Este diseño usa un tipo de diagrama llamado modelo entidad-relación, que muestra las cosas importantes (entidades) y las formas en que están relacionadas. El resultado es diagrama que explica claramente qué datos tendrá la base y cómo se unen, y este diagrama es la base para construir la base de datos relacional después.
- **Diseño lógico:** Su objetivo es tomar el diagrama general de la base de datos y adaptar los tipos de datos y las relaciones entre las tablas, se usa una herramienta llamada

modelo lógico de datos, que organiza la información según el tipo de base que se va a usar, también se revisa que todo esté bien estructurado y sin errores, usando métodos que ayudan a asegurar la calidad del diseño.

- Diseño físico: es la parte final del diseño de una base de datos. Aquí ya se preparan **todas las instrucciones detalladas** para que la base de datos se pueda construir realmente en una computadora, en esta etapa sí se toma en cuenta el **rendimiento**, que la base funcione rápido y de forma eficiente.

2.2 Diccionario de datos

Conecta Magazine (2022) nos dice que un diccionario de datos es una colección centralizada de información sobre los datos, como su significado, relaciones, origen, uso y formato. Actúa como un recurso de referencia para el manejo de bases de datos y sistemas de información, facilitando la comprensión y el uso efectivo de los datos dentro de una organización.

Un diccionario de datos bien cuidado es muy importante para que la información en una organización sea clara, precisa y útil.

Ayuda a que todos los departamentos entiendan y usen los datos de la misma manera, facilita el trabajo en equipo y hace que sea más fácil entender cómo funcionan los sistemas de información.

También es muy útil para tomar decisiones bien informadas, y ayuda a que la empresa cumpla con las reglas sobre el uso correcto de los datos, ya que asegura que todos los usuarios sigan las mismas normas al trabajar con la información.

2.3 Niveles de transparencia

La transparencia se puede entender como la separación de la semántica de alto nivel de un sistema de los aspectos de bajo nivel relacionados a la implementación del mismo, Studocu. (2017).

2.3.1 Transparencia de localización

La transparencia de localización significa que un usuario puede acceder a los datos sin saber en qué computadora o lugar de la red están guardados. La idea es que todo parezca estar en el mismo sitio, aunque por dentro esté repartido en muchas partes.

2.3.2 Transparencia de fragmentación

La transparencia de fragmentación permite que el usuario pueda acceder a la información sin saber si está dividida en partes o fragmentos, ni en dónde están esas partes.

Esto le da al usuario la sensación de que la base de datos está toda en una sola computadora, aunque por dentro esté dividida en varias partes y lugares., el usuario no necesita saber cuántos fragmentos hay, ni sus nombres, ni en dónde están guardados.

2.3.3 Transparencia de réplica

La transparencia de replicación significa que, aunque el sistema tenga copias idénticas de los datos en diferentes computadoras, el usuario no lo nota el cree que los datos son únicos y están en un solo lugar.

El sistema puede crear estas copias para que los datos estén disponibles, por si alguna computadora falla o para que las búsquedas sean más rápidas.

2.4 Fragmentación de datos

La fragmentación se refiere al particionamiento de la información para distribuir cada parte a los diferentes sitios de la red, Clavijero.edu.mx (2025).

2.4.1 Fragmentación horizontal

Este tipo de fragmentación consiste en dividir las filas de una tabla en diferentes grupos, cada grupo se guarda en un servidor distinto y ninguno repite las mismas filas. Lo importante es que todas las filas siguen teniendo las mismas columnas.

2.4.2 Fragmentación Vertical

Aquí lo que se divide no son las filas, sino las columnas de la tabla. Cada fragmento guarda un grupo distinto de columnas y se almacena en un nodo diferente. La única columna que se repite en todos es la clave principal, porque sirve para juntar los datos después si hace falta.

2.4.3 Fragmentación híbrida

Este tipo combina la fragmentación horizontal y vertical, primero se parte la tabla por filas y después cada uno de esos fragmentos se divide por columnas, así terminas con varias partes, cada una con diferentes filas y columnas, según lo que se necesite guardar en cada lugar.

2.5 Distribución de datos

En el Diseño de la Distribución de Datos, lo más importante en concretar es el Diseño de la Fragmentación, mismo del que resultarán definidos los fragmentos de datos que han de colocarse en los distintos sitios del SBDD, Adonai Herrera González (2017).

En el diseño de cómo se van a distribuir los datos en un sistema de base de datos distribuida, lo primero que se hace es el Diseño de la Fragmentación. En esta parte se define cómo se va a dividir la información en fragmentos, y qué datos se van a mandar a cada nodo del sistema

2.5.1 Algoritmos de distribución de datos no replicados

No se llevará a cabo replicación alguna de los fragmentos definidos en el Diseño de la Fragmentación. Es decir, sólo existirá una copia física (el original) de cada fragmento definido, Adonai Herrera González (2017).

En este caso, no se van a hacer copias extras de los fragmentos de la base de datos, cada fragmento que se definió en el diseño tendrá solo una versión original, es decir, una sola copia física guardada en un lugar específico.

2.5.2 Algoritmos de distribución de datos replicados

Habrà múltiples copias (dos o más) de cada fragmento que se haya establecido durante el diseño de la fragmentación.

En este tipo de distribución, cada fragmento de datos tendrá dos o más copias, no solo una. esto significa que la información estará replicada en varios lugares, no concentrada en un solo nodo. Así, si una copia falla o no está disponible, se puede acceder a otra sin problema

3. Procesamiento de consultas distribuidas

Las consultas distribuidas detienen acceso a datos de varios orígenes de datos heterogéneos. Estos orígenes de datos pueden estar almacenados en el mismo equipo o en equipos diferentes. El procesamiento de consultas tiene varias etapas a seguir para resolver una consulta SQL, las características del modelo relacional permiten que cada motor de base de datos elija su propia representación que, comúnmente, resulta ser el álgebra relacional, Tópicos de Bases de Datos (2016).

En un sistema hay otros factores importantes que se deben considerar para que todo funcione bien:

- El costo de transmitir datos por la red: Si los datos están muy dispersos y se consultan frecuentemente, eso puede hacer que el sistema se vuelva más lento o más costoso.
- Repetición y fragmentación: A veces se hacen replicas o se dividen los datos en fragmentos para mejorar el acceso o la organización, pero esto también puede complicar las cosas si no se manejan bien.
- Procesamiento de intersección simple: Esto se refiere a cuando una consulta necesita juntar datos que están en diferentes fragmentos.

3.1 Metodología del procesamiento de consultas distribuidas.

Studocu (2021) nos dice que las consultas distribuidas se encargan de detener el acceso a datos de varios orígenes de datos heterogéneos. Estos orígenes de datos pueden estar almacenados en el mismo equipo o en equipos diferentes

La función principal del procesador de consultas relacionales es convertir una consulta escrita en un lenguaje de alto nivel, se encarga de traducir lo que el usuario quiere obtener en instrucciones que la base de datos pueda seguir para encontrar la información correcta.

Pare ello es importante que exista cierta heterogeneidad en los datos, que puedan provenir de distintas fuentes o estar organizados de diferentes maneras, pero aun así puedan combinarse.

3.2 Estrategias de procesamiento de consultas distribuidas

El procesamiento de consultas tiene varias etapas a seguir para resolver una consulta SQL, las características del modelo relacional permiten que cada motor de base de datos elija su propia representación que, comúnmente, resulta ser el álgebra relacional, Blogspot.com (2025).

3.2.1 Árboles de consulta

Es una estructura de árbol que corresponde a una expresión del álgebra relacional en el que las tablas se representan como nodos hojas y las operaciones del álgebra relacional como nodos intermedios, Studocu (2021).

El árbol de consulta es una forma visual de representar cómo se va a ejecutar una consulta en una base de datos.

- Las hojas (las partes de abajo) representan las relaciones o tablas que se van a usar.
- Las ramas o nodos internos son las operaciones del álgebra relacional, como seleccionar, proyectar o unir datos.

La ejecución del árbol se hace de abajo hacia arriba, se empieza resolviendo las operaciones más simples en las hojas, y el resultado de cada operación se va sustituyendo en el árbol.

3.2.2 Transformaciones equivalentes

Cuando una base de datos está distribuida en varios servidores y conectada a distintos nodos se deben considerar lo siguiente:

1. Un nodo hace una petición al servidor: solicita acceso o consulta a la base de datos.
2. El servidor recibe múltiples solicitudes al mismo tiempo: esto puede generar una sobrecarga si muchos nodos intentan acceder a la misma base de datos local al mismo tiempo.
3. El servidor responde enviando los resultados y asigna un proceso de ejecución (hilo) a cada una de las máquinas nodo dentro de la red local.

Para realizar una transformación en la consulta primero se desfragmenta siguiendo los estándares marcados por las reglas formales y posteriormente se realiza el envío y la máquina que recibe es la que muestra el resultado pertinente para el usuario, de esta se puede producir una copia que será la equivalente a la original, Studocu (2021).

3.2.3 Métodos de ejecución usando la función join

Existen diferentes algoritmos que pueden obtener transformaciones eficientes en el procesamiento de consultas:

➤ **Join externo.**

Este tipo de unión es útil cuando quieres mantener la información, aunque no haya coincidencias entre las tablas.

➤ **Join complejos.**

Los algoritmos de join en bucle anidado y bucle anidado por bloques son soluciones funcionales para unir tablas, aunque no siempre son las más eficientes.

➤ **Join por asociación híbrida.**

Se usa cuando se dispone de una buena cantidad de memoria, aunque no la suficiente como para cargar toda la tabla.

Este algoritmo permite aprovechar mejor los recursos para hacer la unión de forma más eficiente.

➤ **Join por asociación.**

Utiliza una función de asociación (hash) que divide las filas de ambas relaciones en grupos, lo cual hace el proceso más ágil.

➤ **Join por mezcla.**

Consiste en ordenar previamente las dos tablas según las columnas que se van a comparar, y luego recorrerlas al mismo tiempo para encontrar coincidencias.

➤ **Join en bucles anidados por bloques.**

En lugar de comparar fila por fila, este algoritmo procesa las relaciones por bloques de datos, lo que reduce la cantidad de accesos al disco y mejora el rendimiento general.

3.3 Optimización de consultas (global y local)

IBM.com. (2025) argumenta que el objetivo de la optimización global es producir un plan de acceso que optimiza las operaciones de consulta en todos los orígenes de datos globalmente, en todo el sistema federado.

Lo que se busca es que el plan de acceso a los datos sea lo más eficiente posible, cuando se logra un plan eficiente globalmente, significa que se ha encontrado la forma más rápida o económica de ejecutar una consulta, considerando todos los orígenes de datos al mismo tiempo.

Bustos, J. (2025). Nos dice que la optimización local se basa específicamente en condiciones locales dentro de un programa fuente y no se contempla el flujo de ejecución del programa para ese tipo de optimización. Lo que se contempla es particionar el código en bloques básicos de código (alto nivel) y sobre estos fragmentos se lleva a cabo en la optimización.

Se trabaja sobre fragmentos de código en cada nodo, usando su propio esquema local en si permite que se puedan aplicar los algoritmos más adecuados para ejecutar operaciones relacionales dentro de cada nodo. Me parece importante porque ayuda a que el sistema aproveche mejor sus recursos de forma local antes de coordinarse con otros nodos.

4. Manejo de transiciones

Una transacción en un sistema de gestión de bases de datos (SGBD), es un conjunto de ordenes que se ejecutan formando una unidad de trabajo, es decir, una forma indivisible o atómica, itpn (2015).

Es el proceso de realizar una o varias operaciones en una base de datos, que pueden ir desde algo muy simple como consultar un dato, hasta tareas más complejas, como actualizar varios registros al mismo tiempo.

4.1 Estructura de transacciones

La estructura de una transacción usualmente viene dada según el modelo de la transacción, estas pueden ser planas o anidadas, itpn (2015).

- **Transacciones planas:** está formada por una secuencia de operaciones básicas que se ejecutan como una sola unidad, van encerradas entre las palabras clave BEGIN y END, que indican dónde empieza y dónde termina la transacción.
- **Transacciones anidadas:** son situaciones donde una transacción depende de otra, una transacción más grande contiene dentro de sí a otras más pequeñas, esto permite dividir una operación compleja en partes más pequeñas.

4.2 Ejecución de transacciones centralizadas y distribuidas

El procesamiento de bases de datos distribuidas es el procesamiento de bases de datos en el cual la ejecución de transacciones y la recuperación y actualización de los datos acontece a través de dos o más computadoras independientes, por lo general separadas geográficamente, itpn (2015).

4.3 Control de concurrencia

El control de transacciones concurrentes en una base de datos brinda un eficiente desempeño del Sistema de Base de Datos, puesto que permite controlar la ejecución de transacciones que operan en paralelo, accediendo a información compartida, itpn (2015).

Al trabajar en paralelo, estas transacciones pueden acceder a los mismos datos, lo que podría provocar errores si no se gestiona bien, el control de concurrencia se encarga de coordinar estas operaciones, evitando que interfieran entre sí.

4.3.1 Serialización de transacciones

Itpn (2015). Nos plantea que permite el proceso de transacciones asignándoles tiempos de procesamiento el cual permite incrementar el rendimiento del sistema ya que se ejecuta un máximo de procesos en forma concurrente y no a través de una serie.

La ventaja de este sistema es que se pueden hacer dos operaciones al mismo tiempo, lo que mejora la eficiencia y rapidez, pero esto también hace que el proceso para coordinar y sincronizar esas operaciones sea más complicado, porque hay que asegurarse de que no se afecten entre sí y que los datos se mantengan correctos, también es importante contar con algoritmos de control que supervisen como se usan los datos para que todo funcione bien.

4.3.2 Algoritmo de control de concurrencia

Deben sincronizar la ejecución de transacciones concurrentes bajo el criterio de correctitud. La consistencia entre transacciones se garantiza mediante el aislamiento de las mismas, Itpn (2015).

Cuando varias transacciones se ejecutan al mismo tiempo en una base de datos, es fundamental que su ejecución se controle para ello los algoritmos de control de concurrencia se encargan de sincronizar las transacciones, siguiendo reglas que garanticen que el resultado sea correcto como si las transacciones se hubieran hecho una por una.

CAPÍTULO II: METODOLOGÍA

5. Herramientas para las bases de datos distribuidos

5.1 Modelo cliente/servidor

Andrés Schiaffarino. (2019), argumenta que la arquitectura cliente-servidor tiene dos partes claramente diferenciadas, por un lado la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones. En esta arquitectura el cliente suele ser estaciones de trabajo que solicitan varios servicios al servidor, mientras que un servidor es una máquina que actúa como depósito de datos y funciona como un sistema gestor de base de datos, este se encarga de dar la respuesta demandada por el cliente.

Lo más importante de este modelo es que permite que muchos usuarios (clientes) accedan al mismo servidor al mismo tiempo, esto es clave hoy en día, ya que la mayoría de los servicios y aplicaciones deben ser capaces de atender a varios usuarios simultáneamente, como ocurre con redes sociales, correos electrónicos, plataformas de streaming.

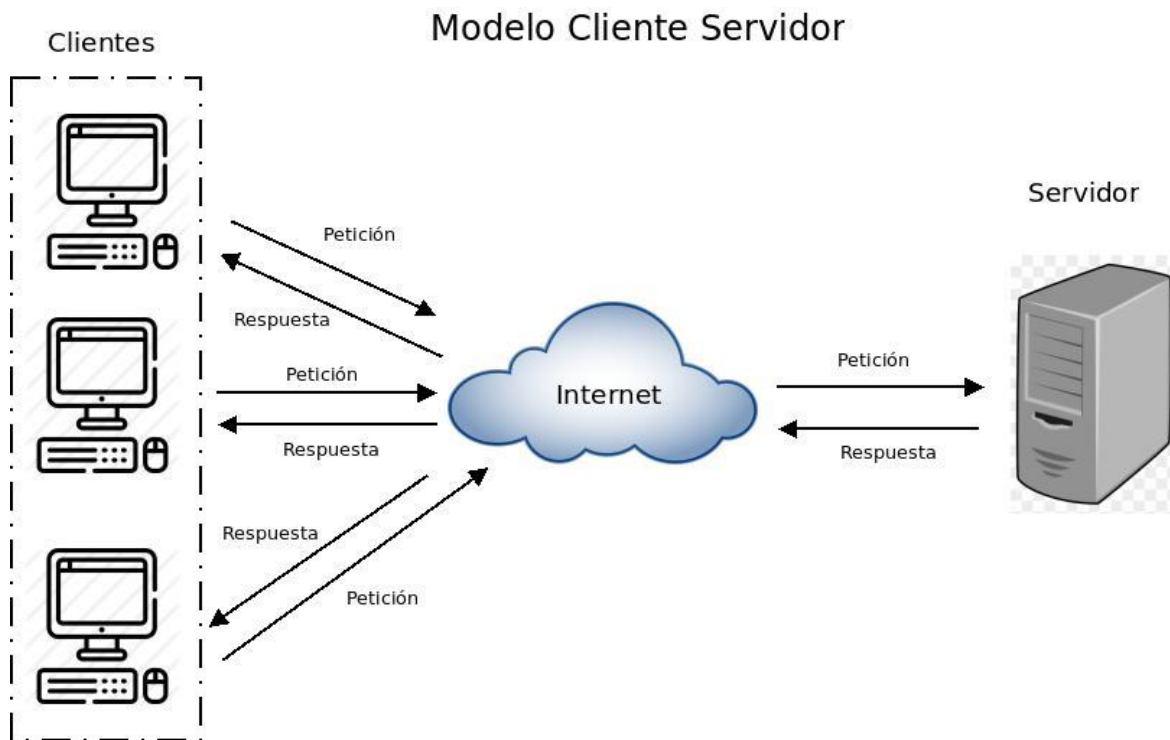


Figura. 3 Esquema del Modelo Cliente Servidor
Obtenido de: <https://blog.infranetworking.com/modelo-cliente-servidor/>

Componentes del modelo cliente-servidor.

- **Red:** Es el conjunto de clientes, servidores y bases de datos conectados entre sí, ya sea de forma física (como con cables) o no física (Wi-Fi).
- **Cliente:** Es el que solicita servicios o información dentro de la red.
- **Servidor:** Es el que ofrece servicios o información a los clientes su función principal es responder a las peticiones que llegan desde los clientes.
- **Protocolo:** Es el conjunto de reglas y pasos que se siguen para enviar y recibir información dentro de la red.
- **Servicios:** Son las respuestas o funciones que ofrece un servidor para satisfacer las solicitudes del cliente.
- **Base de datos:** Son lugares donde se guarda y organiza información.

5.2 Protocolo TCP/IP

5.3 Gestores de base de datos que soportan arquitecturas distribuidas

- Oracle Database: Permite distribuir y acceder a la base de datos desde varios servidores al mismo tiempo, manteniendo alta disponibilidad y escalabilidad.
- Microsoft SQL Server: Soporta alta disponibilidad y puede distribuir datos entre varias instancias del servidor.
- PostgreSQL: puede convertirse en un sistema distribuido.
- MySQL Cluster: permite distribuir tanto datos como nodos en la red, ofreciendo tolerancia a fallos.
- MongoDB: este sistema NoSQL trabaja muy bien con arquitecturas distribuidas gracias los fragmentos que reparte datos entre distintos nodos.

5.4 Infraestructura para un sistema de bases de datos distribuida

Un sistema de base de datos distribuida permite que las aplicaciones accedan a datos de bases de datos locales y remotas. En un sistema de base de datos distribuido homogéneo, cada base de datos es una base de datos Oracle. En un sistema de base de datos distribuido heterogéneo, al menos una de las bases de datos no es una base de datos Oracle. Las bases de datos distribuidas utilizan a cliente/servidor Arquitectura para procesar solicitudes de información, Oracle.com (2025).

- **Sistemas de bases de datos distribuidas homogéneas:** Un sistema de base de datos distribuido homogéneo es básicamente una red que conecta dos o más bases de datos del mismo tipo, lo importante de este tipo de sistema es que una aplicación puede acceder a varias bases de datos al mismo tiempo, como si todo fuera una sola.

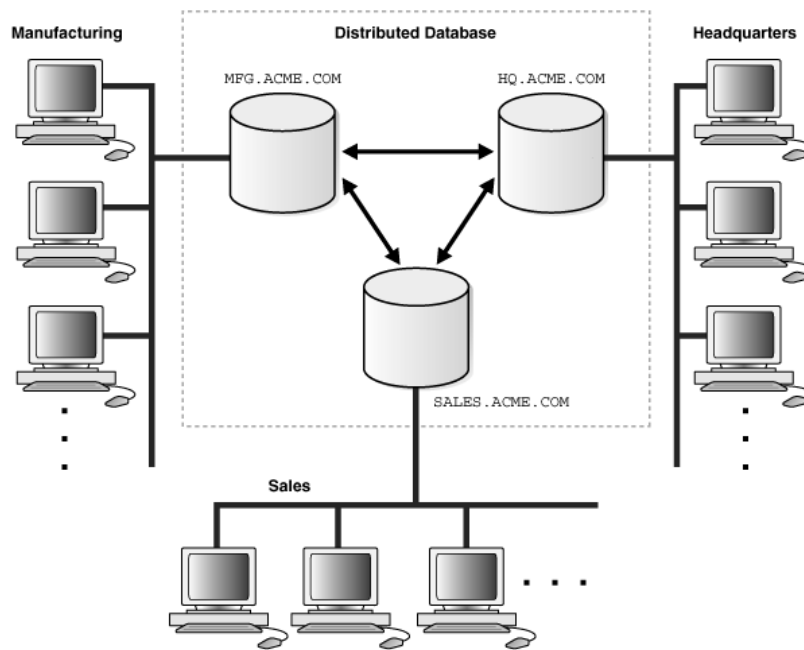


Figura. 4 Base de datos distribuida homogénea

Obtenido de:

https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts001.htm#ADMIN12074

- **Sistemas de bases de datos distribuidas heterogéneas:** es cuando las bases de datos que están conectadas no son del mismo tipo, aunque por detrás estén usando diferentes tecnologías, para la aplicación todo se ve como si fuera una sola base de datos.

5.4.1 Hardware

El hardware que compone una base de datos distribuida se reduce a servidores y la red.

5.4.2 Software

- **Sistema manejador de base de datos distribuida (DDBMS):** Es un conjunto de programas que trabajan en varias computadoras. Su función principal es controlar todo lo relacionado con una base de datos que está repartida en varios lugares.

- Sistema manejador de base de datos (DBMS): Es un programa que se encarga de trabajar con una parte específica de una base de datos distribuida su función es buscar o modificar datos cuando recibe instrucciones del manejador central que coordina todo el sistema.

5.5 MongoDB

MongoDB es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado, MongoDB (2018).

El modelo de documentos de MongoDB es bastante fácil de entender y usar ofrece a los desarrolladores todas las herramientas necesarias para trabajar con aplicaciones, incluso si son muy complejas o si necesitan funcionar a gran escala.

5.6 Docker

Docker. (2025) argumenta que Docker es un paquete de software ejecutable, independiente y liviano que incluye todo lo necesario para ejecutar una aplicación: código, tiempo de ejecución, herramientas del sistema, bibliotecas y configuraciones del sistema.

Las imágenes de contenedores son plantillas que se convierten en contenedores reales cuando se ponen a funcionar, en el caso de Docker, estas imágenes se transforman en contenedores al momento de ejecutarlas, lo bueno de los contenedores es que sirven tanto para aplicaciones en Linux como en Windows, y lo mejor es que siempre se comportan igual, sin importar en qué computadora o servidor se estén ejecutando.

5.7 ReplicaSet

Un ReplicaSet se define con campos, incluyendo un selector que indica cómo identificar a los Pods que puede adquirir, un número de réplicas indicando cuántos Pods debería gestionar, y una plantilla pod especificando los datos de los nuevos Pods que debería crear para conseguir el número de réplicas esperado, Kubernetes (2023, July 25).

Un ReplicaSet en Kubernetes se encarga de asegurarse de que siempre haya una cantidad específica de pods iguales funcionando, si alguno falla o se elimina, ReplicaSet crea uno nuevo automáticamente para mantener el número deseado esto ayuda a que la aplicación siempre esté disponible y funcionando como se espera.

CONCLUSIONES

BIBLIOGRAFÍAS

1. Chandima Jayamina. (2024, June 3). Distributed Database Concepts (Overview) - Chandima Jayamina - Medium. Medium. <https://chandimajayamina.medium.com/distributed-database-concepts-overview-d6c4e85fb4b5>
2. 1.2. Diseño de base de datos distribuidas. (2016, December). Tópicos de Bases de Datos. <https://topicdb.wordpress.com/1-2-diseno-de-base-de-datos-distribuidas-4/>
3. Diccionario de datos. (2022, July 19). Conecta Magazine. <https://www.conectasoftware.com/magazine/glosario/diccionario-de-datos/>
4. Studocu. (2017). Studocu. <https://www.studocu.com/latam/document/universidad-pedagogica-de-el-salvador/metodos-y-tecnicas-de-investigacion/bases-de-datos-distribuidas/24329505>
5. Tema 3.1 Importancia del diseño de una base de datos distribuidas - Bases de datos distribuidas - Instituto Consorcio Clavijero. (2025). Clavijero.edu.mx. https://cursos.clavijero.edu.mx/cursos/080_bdd/modulo3/contenidos/tema3.1.html?opc=0
6. Diseño de Base de Datos Distribuidas. (2017, February 20). Adonai Herrera González. <https://adonaihg.wordpress.com/2017/02/20/disenio-de-base-de-datos-distribuidas/>
7. 1.4. Procesamiento de consultas distribuidas. (2016, December). Tópicos de Bases de Datos. <https://topicdb.wordpress.com/1-4-procesamiento-de-consultas-distribuidas-2-2/>
8. Studocu. (2021). Studocu. <https://www.studocu.com/es-mx/document/instituto-tecnologico-del-norte-de-nayarit/operaciones-mecanicas/metodologia-de-procesamiento-de-consultas-distribuidas/67998843>

9. Estrategias de procesamiento de consultas distribuidas. (2025). Blogspot.com.
<http://basesdatosdistribuidas.blogspot.com/2012/10/estrategias-de-procesamiento-de.html>
10. Studocu. (2021). Studocu. <https://www.studocu.com/es-mx/document/universidad-mexiquense-del-bicentenario/bases-de-datos-distribuidas/unidad-iii-estrategias-consultas-distribuidas/82507045>
11. **Bases de datos distribuídas (2015).** Actividad #16. Blogspot.com.
<https://bases-datos-hernandez-soto.blogspot.com/2015/04/actividad-16.html>
12. Bustos, J. (2025). T11 - Optimización de Consultas, Optimización Global de Consultas y Optimización Local de Consultas. Scribd.
<https://es.scribd.com/presentation/847364536/T11-Optimizacion-de-Consultas-Optimizacion-Global-de-Consultas-y-Optimizacion-Local-de-Consultas>
13. Db2 for Linux, UNIX and Windows. (2025, June 5). Ibm.com.
<https://www.ibm.com/docs/es/db2/12.1.0?topic=federation-global-optimization>
14. itpn (2015).
<http://www.itpn.mx/recursositcs/5semestre/basededatosdistribuidas/Unidad%20IV.pdf>
15. Andrés Schiaffarino. (2019, March 12). Modelo cliente servidor. Infranetworking.
<https://blog.infranetworking.com/modelo-cliente-servidor/>
16. Distributed Database Architecture. (2025). Oracle.com.
https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts001.htm#ADMIN12074
17. ¿Qué Es MongoDB? (2018). MongoDB.
<https://www.mongodb.com/es/company/what-is-mongodb>
18. What is a Container? | Docker. (2025, May 22). Docker.
<https://www.docker.com/resources/what-container/>
19. ReplicaSet. (2023, July 25). ReplicaSet. Kubernetes.
<https://kubernetes.io/es/docs/concepts/workloads/controllers/replicaset/>
20. Crespo Martínez, L. M., & Candelas Herías, F. A. (1998). *Introducción a TCP/IP: Sistemas de transporte de datos*. Publicaciones de la Universidad de Alicante.

21. Date, C. J. (2001). *Introducción a los sistemas de bases de datos* (7.^a ed., A. Núñez Ramos, Trad.). Pearson Educación. (Obra original publicada en 2000 como *An Introduction to Database Systems*, 7th ed.)
22. **Silberschatz, A., Korth, H. F., & Sudarshan, S.** (2011). *Fundamentos de bases de datos* (6^a ed.). McGraw-Hill.
23. **Ramamritham, K., & Gehrke, J.** (2003). *Bases de datos distribuidas* (Ed. en español). Alfaomega.