

**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

FACULTAD DE CIENCIAS DE LA INGENIERÍA CARRERA DE INGENIERÍA EN SOFTWARE

**TEMA:**

BASES DE DATOS DISTRIBUIDAS

**INTEGRANTES:**

ERICKA ELIZABETH BRAVO MECA

JEANDAVID RODOLFO CABRERA GUERRA

MICHELL SUSAN AVILES LITARDO

**MATERIA:**

APLICACIONES DISTRIBUIDAS

**DOCENTE:**

ING. GLEISTON CICERON GUERRERO ULLOA

QUEVEDO- LOS RÍOS – ECUADOR

2022- 2023

**Contenido**

[**CONCEPTOS** 3](#_Toc132144623)

[**ARQUITECTURA DE LAS BASES DE DATOS** 3](#_Toc132144624)

[**TIPOS DE ALMACENAMIENTO** 5](#_Toc132144625)

[**NIVELES DE TRANSPARENCIA EN UNA BASE DE DATOS DISTRIBUIDA** 12](#_Toc132144626)

[**PROCESAMIENTO DISTRIBUIDO DE CONSULTAS** 12](#_Toc132144627)

[**RECUPERACIÓN** 13](#_Toc132144628)

[**VENTAJAS Y DESVENTAJAS** 13](#_Toc132144629)

[**EJEMPLO REALIZADO EN CLASES** **¡Error! Marcador no definido.**](#_Toc132144630)

[**REFERENCIAS** 40](#_Toc132144631)

# **CONCEPTOS**

Una base de datos distribuida (DDBMS, por sus siglas en inglés) es una base de datos que se extiende sobre varias máquinas físicas, que pueden estar ubicadas en diferentes lugares geográficos y conectadas mediante una red de comunicaciones[1]. Las bases de datos distribuidas se utilizan para resolver problemas de escalabilidad y disponibilidad, así como para permitir la gestión y el acceso a datos en tiempo real en sistemas distribuidos y aplicaciones en la nube[2].

En una base de datos distribuida, los datos se dividen en fragmentos y se distribuyen en diferentes nodos o servidores de la red, lo que permite procesar y almacenar grandes volúmenes de datos[3]. Cada nodo o servidor es responsable de un subconjunto de los datos y proporciona un conjunto de operaciones de base de datos a los usuarios finales[4].

Una característica importante de las bases de datos distribuidas es que deben ser capaces de mantener la consistencia de los datos a través de todos los nodos de la red[5]. Para lograr esto, se utilizan técnicas como la replicación de datos, la sincronización y la coordinación de transacciones.

Además, la seguridad también es una preocupación importante en las bases de datos distribuidas, ya que los datos se almacenan y se transmiten a través de una red de comunicaciones[6]. Se deben tomar medidas para garantizar la seguridad de los datos y protegerlos contra accesos no autorizados y ataques malintencionados.

Diagrama

Descripción generada automáticamente

Ilustración 1 Entorno de base de datos distribuida

# **ARQUITECTURA DE LAS BASES DE DATOS**

Una arquitectura de bases de datos distribuidas se puede definir como un conjunto de componentes de software y hardware que trabajan juntos para almacenar, procesar y gestionar datos de manera distribuida en múltiples nodos de red [7]. En esta arquitectura, cada nodo tiene su propia base de datos local y los datos se distribuyen y replican entre los nodos para mejorar la disponibilidad, la escalabilidad y el rendimiento del sistema en general [8]. La arquitectura de bases de datos distribuidas también puede incluir técnicas para mantener la consistencia de los datos entre los nodos y para garantizar la seguridad y privacidad de los datos en un entorno distribuido [9].

**CLIENTE-SERVIDOR**

La arquitectura cliente-servidor es un modelo informático en el que el servidor aloja, entrega y gestiona la mayoría de los recursos y servicios que serán consumidos por el cliente[10]. Este tipo de arquitectura tiene uno o más equipos cliente conectados a un servidor central a través de una red o conexión a Internet [11]. La arquitectura cliente-servidor también es conocida como un modelo de computación en red o red cliente-servidor, ya que todas las solicitudes y servicios son entregados a través de una red[10].

Se utiliza para distribuir y gestionar el acceso a los recursos y servicios de una aplicación. En este modelo, el software se divide en dos partes principales: el cliente y el servidor [12].

Diagrama

Descripción generada automáticamente

Ilustración 2 Arquitectura cliente/servidor

El cliente es la aplicación o interfaz de usuario que se ejecuta en el equipo del usuario final y que se utiliza para interactuar con la aplicación. El servidor es la parte que se ejecuta en un equipo centralizado y que proporciona los recursos y servicios requeridos por el cliente [12].

La comunicación entre el cliente y el servidor se realiza a través de una red, como Internet, y se lleva a cabo mediante protocolos específicos. En general, el cliente envía solicitudes al servidor y este responde con los recursos y servicios necesarios[13].

**PEER TO PEER**

La arquitectura peer to peer (P2P) es un modelo de computación en el que los recursos y servicios son compartidos entre múltiples dispositivos, sin necesidad de un servidor centralizado. En este modelo, cada dispositivo funciona tanto como cliente como servidor, lo que significa que cada uno puede solicitar y proporcionar recursos y servicios a otros dispositivos en la red [14].

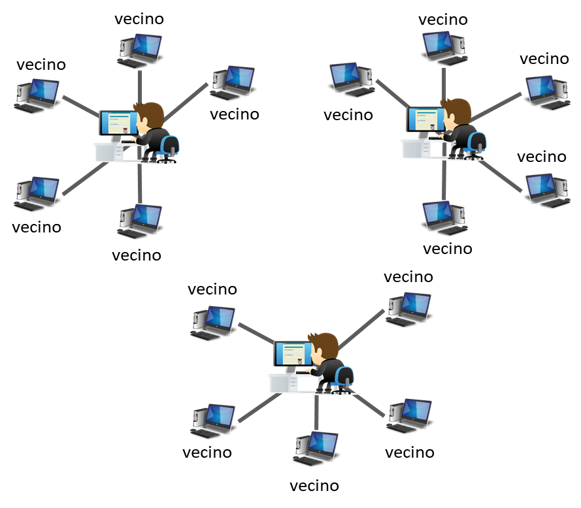


Ilustración 3 Arquitectura peer to peer

En una arquitectura P2P, los dispositivos están conectados en una red descentralizada, en la que no hay un control centralizado ni una jerarquía definida. En su lugar, los dispositivos se comunican directamente entre sí y cooperan para proporcionar los recursos y servicios necesarios [15].

**ARQUITECTURA BASADA EN PARTICIONES**

La arquitectura basada en particiones es una arquitectura de base de datos distribuida que divide una base de datos en varias particiones, cada una de las cuales se almacena en un servidor diferente[16]. Cada partición se gestiona de forma independiente, lo que permite que la base de datos sea escalable horizontalmente[16].

En esta arquitectura, las particiones se pueden dividir de varias maneras, por ejemplo, por rangos, hash o listas[17]. Las particiones se pueden distribuir en diferentes servidores, lo que permite que los datos se almacenen cerca de los usuarios o aplicaciones que los utilizan con mayor frecuencia[18].

Cuando un cliente realiza una solicitud de datos, el servidor correspondiente a la partición en la que se encuentra el dato solicitado maneja la solicitud y devuelve los resultados al cliente[19]. Los servidores coordinan para garantizar la coherencia y la integridad de los datos en todas las particiones[19].

# **TIPOS DE ALMACENAMIENTO**

El almacenamiento en bases de datos distribuidos es una técnica utilizada para mejorar el rendimiento, la disponibilidad y la escalabilidad de una base de datos [20]. En una base de datos distribuida, los datos se almacenan en múltiples servidores en diferentes ubicaciones geográficas, lo que permite procesar grandes cantidades de datos y manejar un gran número de solicitudes simultáneas [20]. Además, si un servidor falla, la base de datos sigue estando disponible debido a la redundancia de los datos almacenados en otros servidores.

La elección del posicionamiento y esquema de almacenamiento de los datos en un sistema distribuido es una de las decisiones más importantes que un diseñador debe tomar. Existen varias opciones para el posicionamiento de los datos, como: almacenamiento, centralizado, replicado, fragmentado o híbrido [21]. Cada una de estas opciones tiene sus propias ventajas y desventajas y es adecuada para diferentes entornos y requisitos de rendimiento y escalabilidad [22].

**CENTRALIZADO**

El almacenamiento de datos centralizado en es una estrategia que se utiliza en algunos casos para mejorar la eficiencia y seguridad de las bases de las bases de datos distribuidas. Es una base de datos que se encuentra almacenada completamente en un único lugar físico, es decir, en una sola máquina y un solo CPU, los usuarios pueden acceder a esta mediante terminales que únicamente muestran los resultados [23].

Diagrama

Descripción generada automáticamente

Ilustración 4 Almacenamiento de datos centralizado

En este caso el almacenamiento centralizado de las bases de datos distribuidas se basa en el clásico modelo servidor/cliente, en el que los clientes se conectan a un servidor centralizado para acceder a los datos de la base de datos [24].

**CARACTERÍSTICAS DEL ALMACENAMIENTO CENTRALIZADO**

Algunas características del almacenamiento de datos centralizado en bases de datos distribuidas son:

* **Un único punto de acceso:** en un sistema de almacenamiento centralizado, todos los datos de la base de datos se almacenan en un único servidor centralizado, que actúa como punto de acceso para los clientes [25].
* **Eficiencia en el acceso a los datos:** como todos los datos están almacenados en un único servidor, el acceso a la información puede ser más rápido y eficiente que en una base de datos distribuida [25].
* **Simplificación de la gestión:** al tener todos los datos en un solo lugar, la gestión y el mantenimiento de la base de datos se pueden simplificar [25].
* **Mayor control y seguridad:** como todos los datos están en un solo lugar, el control y la seguridad de la base de datos pueden ser más fáciles de manejar [26].
* **Mayor capacidad de procesamiento:** al tener todos los recursos de la base de datos en una sola máquina, se puede contar con una mayor capacidad de procesamiento[26].
* **Vulnerabilidad a fallos:** si el servidor central falla, todo el sistema de la base de datos puede verse afectado, lo que hace que este tipo de almacenamiento sea vulnerable a fallos [26].
* **Vulnerabilidad a ataques:** como la base de datos se encuentra en un único lugar, puede ser vulnerable a ataques por parte de personas no autorizadas [26].

**VENTAJAS DEL ALMACENAMIENTO CENTRALIZADO**

A continuación, se presenta algunas ventajas del almacenamiento de datos centralizado en bases de datos distribuidas:

* **Se evita la redundancia:** en sistemas que no usan bases de datos centralizadas, cada aplicación tiene sus propios archivos privados o se encuentran en diferentes localidades. Esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento [23].
* **Se evita la inconsistencia:** ya que, si un hecho específico se representa por una sola entrada, la no-concordancia de datos no puede ocurrir [23].
* **Puede conservarse la integridad:** Que los datos se encuentren en una sola máquina ayuda a evitar la inconsistencia de los datos, por el mismo hecho de encontrarse en una sola máquina [23].

**DESVENTAJAS DEL ALMACENAMIENTO CENTRALIZADO**

A continuación, se presenta algunas desventajas del almacenamiento de datos centralizado en base de datos distribuidas:

* Cuando un sistema de base de datos centralizada falla, se pierde toda la disponibilidad de procesamiento y sobre todo de la información confiada al sistema [23].
* Las cargas de trabajo no se pueden difundir entre diferentes computadoras, ya que los trabajos siempre se ejecutarán en la misma máquina [23].
* Los mainframes (computadora central) no ofrecen mejor proporción precio/rendimiento que los microprocesadores de los sistemas distribuidos [23].

**REPLICADO**

En el almacenamiento en bases de datos distribuidas se almacenan los datos en múltiples servidores. Cada servidor tiene una copia completa de la base de datos y los datos se sincronizan automáticamente entre los servidores para garantizar que todas las copias sean consistentes [27].

Diagrama

Descripción generada automáticamente

Ilustración 5 Replicación de datos

Este método implica altos costos para mantener la información, ya que cada cambio hecho en una copia debe replicarse en todas las demás. Por lo tanto, este sistema es muy adecuado para bases de datos en las que se realizan pocas operaciones de escritura, pero muchas de lectura [27].

**CARACTERÍSTICAS DEL ALMACENAMIENTO REPLICADO**

Algunas características del almacenamiento de datos replicado en bases de datos distribuidas son:

* **Redundancia:** Los datos se almacenan en múltiples lugares o sitios para aumentar la disponibilidad y la tolerancia a fallos [28].
* **Escalabilidad:** El almacenamiento replicado permite una mayor escalabilidad y capacidad de procesamiento, ya que se pueden realizar consultas y operaciones en varias copias de los datos simultáneamente [28].
* **Consistencia:** Los datos replicados deben mantenerse consistentes en todos los sitios para evitar inconsistencias o conflictos de datos [28].
* **Confiabilidad:** La replicación de datos aumenta la fiabilidad de la base de datos, ya que, si un sitio falla o se desconecta, los datos aún estarán disponibles en otros sitios [29].
* **Complejidad:** La replicación de datos aumenta la complejidad de la gestión de la base de datos, ya que los cambios realizados en un sitio deben ser propagados a todos los demás sitios [29].

**VENTAJAS DEL ALMACENAMIENTO CENTRALIZADO**

A continuación, se presenta algunas ventajas del almacenamiento de datos replicado en bases de datos distribuidas:

* **Disponibilidad:** El sistema sigue funcionando incluso en caso de caída de uno de los nodos, ya que los datos están disponibles en otros nodos [30].
* **Aumento del paralelismo:** Varios nodos pueden realizar consultas en paralelo sobre la misma tabla. Cuantas más réplicas existan de la tabla, mayor será la posibilidad de que el dato buscado se encuentre en el nodo desde el que se realiza la consulta, lo que minimiza el tráfico de datos entre nodos y aumenta la eficiencia en la ejecución de consultas [30].

**FRAGMENTADO**

La fragmentación de datos es el proceso de dividir una tabla grande en varias partes más pequeñas llamadas fragmentos. Cada fragmento de la tabla se almacena en un nodo diferente en la red [31]. La información de fragmentación de datos se almacena en el catálogo de datos distribuido para acceder a ella mediante el procesador de transacciones para procesar las solicitudes de usuario [21]. Existen tres tipos de estrategias de fragmentación: fragmentación horizontal, vertical y mixta.

**FRAGMENTACIÓN HORIZONTAL**

La fragmentación horizontal divide una tabla en subconjuntos de tuplas (filas) basados en la información de la base de datos y la información de la aplicación. Cada fragmento consiste en filas únicas y se almacena en un sitio diferente [27]. La ventaja de la fragmentación horizontal es que permite la localización de los datos almacenando los fragmentos en los sitios donde se acceden con más frecuencia [31]. Se permite el procesamiento en paralelo en los fragmentos de una relación. Un ejemplo de fragmentación horizontal puede ser que un banco fragmente su tabla de clientes por ubicación. Por ejemplo, una tabla de clientes puede dividirse en dos fragmentos, uno con el ID del cliente, nombre y dirección y otro con el ID del cliente y otra información financiera [31].

**EJEMPLO:**

A continuación, se presenta una tabla inicial de alumnos en la cual aplicaremos la fragmentación horizontal.

**Tabla 1 Datos iniciales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Nota ingreso** | **Beca** |
| 87633483 | EUI | Concha Queta | 5.7 | NO |
| 99855743 | EUI | Katherin Baren | 5.8 | NO |
| 33887993 | EUIT | Amarilis Tarira | 7.4 | SI |
| 05399075 | EUI | Luis Vera | 6.1 | NO |
| 44343234 | EUIT | Ana Vera | 10 | SI |
| 85123564 | EUIT | Luis Toapanta | 9.0 | SI |
| 78952134 | EUI | Mauricio Vera | 8.0 | SI |

Como resultado, tenemos dos tablas fragmentadas que se muestran a continuación:

**Tabla 2 Primera tabla fragmentada horizontalmente**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Nota ingreso** | **Beca** |
| 87633483 | EUI | Concha Queta | 5.7 | NO |
| 99855743 | EUI | Katherin Baren | 5.8 | NO |
| 78952134 | EUI | Mauricio Vera | 8.0 | SI |
| 05399075 | EUI | Luis Vera | 6.1 | NO |

**Tabla 3 Segunda tabla fragmentada horizontalmente**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Nota ingreso** | **Beca** |
| 44343234 | EUIT | Ana Vera | 10 | SI |
| 85123564 | EUIT | Luis Toapanta | 9.0 | SI |
| 33887993 | EUIT | Amarilis Tarira | 7.4 | SI |

**FRAGMENTACIÓN VERTICAL**

La fragmentación vertical se puede lograr mediante dos estrategias: agrupación y división. La primera crea un fragmento para cada atributo y los agrupa para construir fragmentos. Mientras que la segunda utiliza un enfoque de arriba hacia abajo que divide progresivamente las relaciones globales en fragmentos [31]. La fragmentación vertical es más compleja que la fragmentación horizontal y se ha utilizado tanto en entornos centralizados como distribuidos. Divide una tabla en subconjuntos de atributos (columnas). Por lo tanto, es necesario definir subconjuntos de atributos que se acceden conjuntamente con frecuencia para admitir el acceso y transferencia de datos eficientes [25]. Cada fragmento se encuentra en un sitio diferente y los atributos únicos, excepto que la columna clave debe estar en cada uno. La fragmentación mixta es la combinación de las estrategias horizontal y vertical. Una tabla se puede dividir en subconjuntos de filas, cada una con un subconjunto de columnas. Utilizando el mismo ejemplo, un banco puede particionar su tabla de clientes por ubicación y agruparla por ciertos atributos [31].

**FRAGMENTACIÓN MIXTA**

La fragmentación mixta es una técnica de fragmentación de bases de datos que combina la fragmentación horizontal y vertical. En la fragmentación mixta, una tabla se divide en subconjuntos de filas, cada uno con un subconjunto de columnas [27]. Esto permite una mayor flexibilidad en la forma en que se almacenan y acceden los datos, lo que puede mejorar la eficiencia y el rendimiento de las consultas. La fragmentación mixta se utiliza a menudo en entornos de bases de datos distribuidas donde los datos se almacenan en múltiples sitios y se acceden a través de una red [31].

Todos los algoritmos de fragmentación deben cumplir tres criterios de corrección:

* **Completitud:** La fragmentación de una relación es completa si y solo si cada elemento de datos en la relación aparece en al menos un fragmento. Este criterio asegura que no haya pérdida de datos durante la fragmentación [31].
* **Reconstrucción:** Debe ser posible utilizar una operación relacional para reconstruir la relación original y asegurar la preservación de las dependencias funcionales [31].
* **Disyunción:** Cada elemento de datos se encuentra en un solo fragmento para minimizar la redundancia [31].

**EJEMPLO:**

**Tabla 4 Datos iniciales**

|  |  |  |  |
| --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Beca** |
| 87633483 | EUI | Concha Queta | NO |
| 99855743 | EUI | Katherin Baren | NO |
| 33887993 | EUIT | Amarilis Tarira | SI |
| 05399075 | EUI | Luis Vera | NO |
| 44343234 | EUIT | Ana Vera | SI |
| 85123564 | EUI | Luis Toapanta | SI |
| 78952134 | EUI | Mauricio Vera | SI |

**Tabla 5 Primera tabla fragmentada mixta**

|  |  |  |  |
| --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Beca** |
| 87633483 | EUI | Concha Queta | NO |
| 99855743 | EUI | Katherin Baren | NO |
| 33887993 | EUIT | Amarilis Tarira | SI |
| 05399075 | EUI | Luis Vera | NO |
| 44343234 | EUIT | Ana Vera | SI |
| 85123564 | EUI | Luis Toapanta | SI |
| 78952134 | EUI | Mauricio Vera | SI |

**Tabla 6 Segunda tabla fragmentada mixta**

|  |  |  |  |
| --- | --- | --- | --- |
| **DNI** | **Escuela** | **Nombre** | **Nota ingreso** |
| 87633483 | EUI | Concha Queta | 5.7 |
| 99855743 | EUI | Katherin Baren | 5.8 |
| 33887993 | EUIT | Amarilis Tarira | 7.4 |

**CARACTERÍSTICAS DEL ALMACENAMIENTO FRAGMENTADO**

Algunas características del almacenamiento de datos fragmentado en bases de datos distribuidas son:

* **Divide grandes tablas:** la fragmentación se utiliza para dividir grandes tablas en partes más pequeñas y manejables [25].
* **Distribución geográfica:** la fragmentación permite distribuir los fragmentos de datos en diferentes ubicaciones geográficas, lo que puede mejorar el rendimiento y la disponibilidad de los datos [25].
* **Concurrencia:** la fragmentación también puede mejorar la concurrencia de la base de datos, ya que permite la ejecución simultánea de varias transacciones [25].
* **Reducción del tráfico de red:** la fragmentación también puede reducir el tráfico de red, ya que solo se accede a los fragmentos de datos necesarios en cada sitio [32].
* **Mayor seguridad:** la fragmentación también puede mejorar la seguridad de los datos, ya que cada fragmento se puede asignar a un sitio específico y protegerse de manera adecuada [32].

**VENTAJAS DEL ALMACENAMIENTO FRAGMENTADO**

A continuación, se presenta algunas ventajas del almacenamiento de datos fragmentado en bases de datos distribuidas:

* Aumenta el nivel de concurrencia, también denominada, concurrencia interconsulta [27].
* Dado que los datos se almacenan cerca del lugar de uso, aumenta la eficiencia del sistema de base de datos [27].
* Las técnicas de optimización de consultas locales son suficientes para la mayoría de las consultas, ya que los datos están disponibles localmente [27].
* Dado que los datos se almacenan cerca del lugar de uso, la eficiencia del sistema de base de datos se incrementa [32].
* Las técnicas de optimización de consultas locales son suficientes para la mayoría de las consultas, ya que los datos están disponibles localmente [32].
* Dado que los datos irrelevantes no están disponibles en los sitios, la seguridad y la privacidad de la del sistema de bases de datos [32].

**DESVENTAJAS DEL ALMACENAMIENTO FRAGMENTADO**

A continuación, se presenta algunas desventajas del almacenamiento de datos fragmentado en bases de datos distribuidas:

* Cuando se necesitan datos de distintos fragmentos, las velocidades de acceso pueden ser muy elevadas [32].
* En caso de fragmentaciones recursivas, el trabajo de reconstrucción necesitará técnicas costosas [32].
* La falta de copias de seguridad de los datos en distintos sitios puede hacer que la base de datos ineficaz en caso de fallo de un sitio [32].

# **NIVELES DE TRANSPARENCIA EN UNA BASE DE DATOS DISTRIBUIDA**

La transparencia en una base de datos distribuida se refiere a la capacidad de los usuarios para acceder y manipular los datos sin necesidad de conocer su ubicación física o su estructura interna[33]. Hay varios niveles de transparencia que se pueden implementar en una base de datos distribuida, como se describe a continuación:

* **Transparencia de acceso:** Los usuarios pueden acceder a los datos como si estuvieran almacenados localmente, sin conocer la ubicación física de los mismos en la red[34].
* **Transparencia de fragmentación:** Los datos se pueden dividir en fragmentos que se almacenan en diferentes nodos de la red. Los usuarios pueden acceder a los datos como si estuvieran almacenados en una única base de datos[35].
* **Transparencia de replicación:** Los datos se pueden replicar en diferentes nodos de la red para mejorar la disponibilidad y la redundancia. Los usuarios pueden acceder a los datos como si estuvieran almacenados en un solo lugar[36].
* **Transparencia de ubicación:** Los usuarios pueden acceder a los datos sin conocer la ubicación física de los mismos en la red[36].
* **Transparencia de concurrencia:** Los usuarios pueden acceder a los datos sin preocuparse por las interferencias o los bloqueos que puedan ocurrir debido a la concurrencia de múltiples transacciones[37].
* **Transparencia de recuperación:** Los usuarios pueden acceder a los datos sin preocuparse por la recuperación de fallos que puedan ocurrir en la red[37].

# **PROCESAMIENTO DISTRIBUIDO DE CONSULTAS**

El sistema debe ser capaz de procesar consultas que afecten a datos de más de un sitio y hacerlo de forma optimizada. Este hecho puede ser considerado como otra razón por la que los sistemas distribuidos siempre son relacionales (las peticiones relacionales son optimizables, mientras que las no relacionales no lo son)[38].

El procesamiento de consultas distribuidas es una técnica utilizada en los sistemas de bases de datos distribuidas para procesar las consultas que involucran datos que se encuentran en varios nodos del sistema[39]. El procesamiento de consultas distribuidas implica la división de la consulta en subconsultas que se ejecutan en diferentes nodos de la red, y la combinación de los resultados de las subconsultas para obtener el resultado final de la consulta[40].

Algunas características de los procesamientos de consulta distribuidos son:

* **Fragmentación de datos**: el procesamiento de consulta distribuido implica la fragmentación de datos, que es la división de los datos de una tabla en varios nodos del sistema distribuido. La fragmentación puede ser horizontal (basada en filas) o vertical (basada en columnas)[2].
* **Transparencia:** la transparencia es una característica importante del procesamiento de consulta distribuido, que permite que el sistema parezca un sistema centralizado para los usuarios, aunque esté distribuido. Esto se logra a través de la transparencia de ubicación, la transparencia de replicación y la transparencia de fragmentación[41].
* **Optimización de consultas:** el procesamiento de consulta distribuido requiere una optimización cuidadosa de las consultas para minimizar la cantidad de datos que deben transferirse entre los nodos y reducir el tiempo de respuesta de la consulta. La optimización de consultas se puede hacer a nivel local o global[42].
* **Planificación de transacciones:** el procesamiento de consulta distribuido implica la coordinación de las transacciones distribuidas, lo que significa garantizar la integridad y la consistencia de los datos en todos los nodos. La planificación de transacciones se puede hacer a través de un protocolo de control de concurrencia distribuido, como el protocolo de dos fases.
* **Gestión de la concurrencia:** la gestión de la concurrencia es una tarea crítica en el procesamiento de consulta distribuido, ya que varios usuarios pueden acceder a los mismos datos simultáneamente. Esto requiere una estrategia para resolver conflictos de acceso concurrente, como bloqueo de registros o control de versiones[44].

# **RECUPERACIÓN**

La recuperación en bases de datos distribuidas se refiere a la capacidad del sistema de recuperarse de errores y fallos que puedan ocurrir en cualquiera de los nodos que conforman la red[45]. Es decir, el objetivo de la recuperación es garantizar que los datos no se pierdan y que el sistema siga siendo capaz de funcionar correctamente después de un fallo[46].

Existen diferentes técnicas y estrategias para la recuperación en bases de datos distribuidas, como la replicación de datos, la realización de checkpoints, la detección y corrección de errores, entre otras[47]. La elección de la estrategia adecuada depende del tipo de aplicación y de los requisitos de recuperación que se deseen cumplir[48].

En general, la recuperación en bases de datos distribuidas es un tema complejo debido a la necesidad de coordinar las acciones de los nodos distribuidos y garantizar la coherencia de los datos [49]. Sin embargo, existen muchas investigaciones y avances en este campo para mejorar la eficiencia y efectividad de los sistemas de recuperación.

# **VENTAJAS Y DESVENTAJAS**

**Ventajas**

* **Mayor disponibilidad:** En una base de datos distribuida, los datos están almacenados en múltiples servidores, lo que significa que, si uno de los servidores falla, los datos aún estarán disponibles en los otros servidores[50].
* **Mayor escalabilidad:** Las bases de datos distribuidas pueden manejar grandes cantidades de datos y un gran número de usuarios simultáneamente, lo que las hace más escalables que las bases de datos centralizadas[51].
* **Mayor rendimiento:** Al distribuir los datos en múltiples servidores, las bases de datos distribuidas pueden mejorar el rendimiento en comparación con las bases de datos centralizadas[52].
* **Mayor flexibilidad:** Las bases de datos distribuidas permiten a las empresas distribuir sus datos en diferentes ubicaciones geográficas, lo que puede ser beneficioso para las empresas que tienen múltiples oficinas o sucursales en diferentes partes del mundo[53].

**Desventajas**

* **Mayor complejidad:** Las bases de datos distribuidas son más complejas de diseñar, implementar y administrar que las bases de datos centralizadas[54].
* **Mayor costo:** Debido a su complejidad, las bases de datos distribuidas pueden ser más costosas de implementar y mantener que las bases de datos centralizadas[55].
* **Mayor vulnerabilidad:** Las bases de datos distribuidas son más vulnerables a los problemas de seguridad, ya que los datos se almacenan en múltiples servidores y es posible que no todos los servidores tengan los mismos niveles de seguridad[56].
* **Mayor dificultad en la sincronización:** Cuando se realizan cambios en los datos en uno de los servidores, es posible que sea necesario sincronizar esos cambios en todos los otros servidores, lo que puede ser difícil y llevar tiempo[57].

**Practica Base de datos distribuida**

1. Como primer paso debemos desactivar el Firewall de Windows. Con esto lo que haremos será permitir que otros dispositivos puedan conectarse a nuestra red. Esto importante ya que en la realización de nuestra práctica debemos crear una red local para que varios ordenadores se puedan conectar.

* Ingresamos al buscador de Windows
* Buscamos “Firewall de Windows defender”
* Una vez dentro vamos a la opción del lado izquierdo que dice “Activar o desactivar firewall”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 1 Interfaz de inicio de Firewall de Windows defender

* En este apartado buscamos la opción de “Activar o desactivar el Firewall de Windows Defender”, una vez que damos clic nos mostrará las opciones que se pueden observar en la Figura 2. Seleccionamos el radio Button con la opción que diga “Desactivar firewall de Windows Defender”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 2 Opciones de configuración para cada tipo de red

1. Para el siguiente paso realizaremos un levantamiento de servicios de la máquina con ayuda de la opción “Servicios” que se puede encontrar buscando en la barra herramientas administrativas de Windows, los servicios que se tenemos que habilitar son:

* Administrador de Conexión automática de Acceso Remoto
* Administrador de Conexión de Acceso Remoto
* SQL Server Agent
* Conexiones de Red
* Coordinador de transacciones distribuidas
* Enrutamiento y Acceso remoto
* KTMR para DTC (Coordinador de Transacciones distribuidas)
* Registro remoto
* SQL Server Browser

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 3 Interfaz de las herramientas administrativas de Windows

* En las opciones presentadas en la Figura 3 se puede observar todas las herramientas administrativas con las que cuentas Windows. Para realizar el paso de levantamiento de servicios damos clic en la opción Servicios y se nos abrirá una nueva ventana donde buscaremos los servicios antes mencionados ubicando su tipo de inicio en “automático”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 4 Ejemplo de una de las propiedades de los servicios

1. Como siguiente paso abrimos el SQL Server Management Studio Management Studio 19.
   * Ingresamos con la autenticación de SQL Server Authentication, con el usuario sa y la contraseña.
   * En caso de no haber habilitado el usuario sa ingresamos con la opción Windows Authentication.
   * En las opciones de Object Explorer desplegamos la carpeta Segurity, luego la carpeta login, buscamos el usuario sa, damos clic derecho y seleccionamos la opción propiedades.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Figura 5 Usuarios disponibles del SQL Server

* En las opciones nos situamos en general y deshabilitamos la opción Enfoque password policy y escribimos la nueva contraseña

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Figura 6 Opciones de cambio de contraseña

* A continuación, damos clic en la opción status y en el apartado de Login habilitamos esa opción, como se puede observar en la Figura 7.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Figura 7 Opciones de Status

* Una vez que aplicamos los cambios reiniciamos el SQL Server con ayuda de opción Restart.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 8 Reinicio del SQL Server

* A continuación, creamos un nuevo seguimos el mismo recorrido que se utilizó para habilitar el usuario sa, pero en esta ocasión vamos a dar clic en new login, rellenamos los campos requeridos para la creación del nuevo usuario.
* En la opción de sever roles seleccionaremos los roles que se pueden visualizar en la imagen 10 y aplicamos los cambios.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 9 Creación del nuevo usuario

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 10 Server Roles

* Nos desconectamos y volvemos a ingresar, pero ahora con el usuario y contraseña que creamos. Tal como se muestra en la Figura 11.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 11 Inicio de sesión con el usuario creado

* Activamos el acceso DTC de la red para ello buscamos el ejecutar y escribimos el comando dcomcnfg para abrir los servicios de componentes. Desplegamos la opción Servicios de componentes como se puede observar en la Figura 13. En el DTC local abrimos sus propiedades y seleccionamos todas las opciones que se presentan en la Figura 14 en la opción de seguridad.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 12 Comando dcomcnfg

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 13 Servicios de componentes

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 14 Propiedades de DTC local

1. A continuación, vamos a crear una publicación en SQL server, esto debemos realizarlo desde el computador que está como cliente. Recordemos que una publicación permite especificar más fácilmente un conjunto de objetos y datos de bases de datos relacionados lógicamente, que se replican como una unidad.
   * Debemos tener en cuenta que para realizar una publicación debemos tener activado el agente de SQL.
   * También debemos tener instalada la herramienta de replicación, esto lo podemos hacer desde el SQL Server install.
   * Una vez realizado estos pasos con anterioridad, procedemos a crear una publicación, dando clic derecho sobre la opción de “Local publications”.

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Figura 15 Paso para crear nueva publicación SQL

* En esta interfaz le damos a Next, para continuar con la creación de nuestra publicación.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 16 Interfaz nueva publicación.

* Seleccionamos esa opción y le damos a Next, para visualizar la siguiente interfaz.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 17 Inicialización del Agente de SQL Server.

* Seguimos marcando la opción Next, dejando todo predeterminado.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 18 Interfaz Snapshot Folder.

* En este ejemplo solo hemos cargado una base de datos de prueba que es la NORTHWIND, en caso de existir más, simplemente deben elegir la base de datos con la que se va a realizar la publicación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 19 Interfaz publication database.

* En el tipo de publicación debemos elegir para este ejemplo la opción de “Merge publication”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 20 Interfaz publication Type.

* Elegimos la opción de SQL Server 2008 en adelante, ya que tenemos la versión de SQL18.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 21 Interfaz Subscriber Types.

* Luego en la sección de artículos, elegimos los artículos que queremos publicar, en este caso estaremos haciéndolo con las tablas y las vistas de nuestra base de datos.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 22 Interfaz de Articles.

* En esta interfaz, simplemente marcamos Next para poder continuar con la publicación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 23 Interfaz Articles Issues.

* Le damos a continuar y seguimos con el proceso de publicación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 24 Interfaz Fiter Table Rows

* Continuamos a la siguiente interfaz dejando todo predeterminado como lo ofrece SQL Server.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 25 Interfaz Snapshot Agent.

* Dentro de esta interfaz seleccionamos la opción de Security Settings, para agregar la configuración de la publicación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 26 Interfaz Agent Security.

* En el primer recuadro marcamos la opción de la parte de abajo, y en el segundo indicamos que vamos a usar un inicio de sesión con una cuenta de SQL y aquí ubicamos los datos de nuestro servidor. Luego damos clic en OK.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 27 Interfaz Snapshot Agent Security.

* En esta pestaña marcamos la opción Next, para dar por culminada la creación de la publicación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 28 Interfaz Wizar Actions.

* Se nos muestran detalles de todas las configuraciones realizadas en nuestra publicación, una vez que estemos de acuerdo le damos a finalizar y la publicación procederá a crearse.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 29 Interfaz Complete the Wizar.

* Una vez que haya sido creada correctamente deberá mostrase así, en caso de que salgan 4 opciones y 2 de estas sean alertas, deben verificar que el agente de SQL Server este activado y volver a realizar el proceso, hasta que se muestre como la siguiente imagen.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 30 Interfaz final de la creación de una publicación.

1. A continuación, vamos a crear una nueva suscripción. Recordemos que la suscripción sirve para definir qué datos de publicación, se recibirán, dónde y cuándo.
   * Nos situamos en Local Subcriptions, damos clic derecho y elegimos crear una nueva suscripción.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 31 Creación de una nueva suscripción.

* Nos aparecerá esta interfaz inicial, debemos darle a Next para iniciar con la creación de una nueva suscripción.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 32 Interfaz inicial para crear una nueva suscripción.

* En esta interfaz que se muestra en la figura debemos elegir la base de datos y la publicación que hemos creado anteriormente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 33 Interfaz de la publicación.

* Dejamos como nos aparece predeterminado y le damos a Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 34 Interfaz de Merge Agent Location.

* Aquí nos aparece la instancia que tenemos enlazada para realizar nuestra suscripción, para agregar una nueva debemos darle en la parte de add SQL server Subscriber.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 35 Interfaz para agregar suscripciones SQL.

* Cuando le demos clic a esa opción nos aparecerán las maquinas que están conectadas a nuestra red. Para tener conectadas varias máquinas a nuestra misma red podemos hacer uso del software llamada Hamachi. A continuación, se muestra el proceso para tener dos máquinas conectadas a una misma red.
* Luego de descargar Hamachi de la página oficial, y de iniciar sesión. Se procede a crear una nueva red.
* Para esto nos dirigimos a la opción de red, que se encuentra en la parte superior y damos clic en “crear una red”.
* Luego llenamos la información creando un nombre a la red y luego una contraseña para realizar la conexión.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Para realizar la conexión desde el otro computador, nos dirigimos a la opción de red, que se encuentra en la parte superior y damos clic en “unirse a una red”.
* Luego llenamos con los datos proporcionados por el ordenador que creó la red.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Visualizamos la conexión de las dos máquinas a la misma red, además de sus direcciones IP, que estaremos usando para realizar la conexión en SQL.

Captura de pantalla de un celular

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Una vez realizado este proceso, podemos encontrar en nuestra red de servidores a la otra maquina con la que queremos realizar la suscripción de publicación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* En esta sección vemos como se ha agregado correctamente la instancia que hemos indicado anteriormente, verificamos que si inicialice inmediatamente y le damos a Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Luego de haber verificado el punto anterior, tenemos que indicar que esta suscripción se está corriendo continuamente. Y le damos a Next.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* En esta interfaz, tenemos que indicar a que servidor vamos a realizar la suscripción. Para esto vamos a dar clic en la opción de connection to Subscriber.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Indicamos la segunda opción en el primer recuadro, y en el último al igual que el proceso de publicación tenemos que indicar el usuario donde está el servidor con las credenciales.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Luego de que agregamos los datos correspondientes de la conexión, tenemos que indicar el tipo de suscripción que en este caso estamos como clientes, ya que nos estamos conectando a un servidor. Le damos clic en Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Luego esperamos y comprobamos de que no haya ningún error y finalizamos la suscripción.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Conexión de dos servidores en SQL Server.

* Como primer paso ingresamos con los usuarios que vamos a realizar la conexión, en este caso PC1 “USUARIO”, PC2 “sa”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Luego nos dirigimos a la sección de “Server Objects” – “Linked Servers”, y damos clic en “New Linked Server”.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Luego ubicamos la dirección IP, o el nombre de la máquina con la que queremos hacer link. En caso de tener una instancia con otro nombre debe ubicarse, si no le mostrará error. Debe indicar la opción “SQL Server”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Nos dirigimos a la sección de seguridad, y marcamos la última opción para indicarle con que usuario voy a realizar esa conexión. Luego ingreso las credenciales correctas para poder hacer link.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* En el otro ordenador sería de esta forma.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Le damos en Ok, y luego de realizar la conexión nos debería aparecer de esta forma.

Imagen que contiene Tabla

Descripción generada automáticamente

* Podemos ver que se muestra la IP, del otro ordenador y dentro de la misma, tenemos acceso a la base de datos de NORTHWIND. A continuación, vamos a realizar una consulta para ver los registros que están dentro de la base de datos.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

* Esta consulta es generada por SQL, dando clic derecho sobre la tabla y generando un script de la instrucción select.

1. Realizando consultas (select, insert, update, delete).

* Realizaremos las consultas en el ordenador PC1, y las podremos ver aplicadas de forma inmediata en el ordenador PC2.
* Aquí se ve cumplido que se pueden realizar cambios en el ordenador principal y estos van a estar reflejados en los demás ordenadores.

Una captura de pantalla de una computadora

Descripción generada automáticamente

* Veamos los resultados, en el ordenador 2. Actualización del registro número 4.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

* Se terminó eliminando el registro número 78, por los problemas que presentaban el registro número 2, ya que estaba enlazado con otras tablas.
* A continuación, se muestra el eliminación e inserción de un registro.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

1. Conclusión de la práctica: El proceso de la práctica ha sido algo tedioso. Primeramente, se ha tenido que visualizar los videos que han sido publicados y no poseían audio, por lo que se tuvo que ir sobre la marcha dándonos cuenta de los detalles para saber que configuraciones debía realizar la máquina cliente y la máquina servidor.

También se omitieron algunos puntos importantes, ya que los practicantes del video habían realizado el proceso con anterioridad y omitieron algunas configuraciones, y eso sin contar con los errores que se presentaron poseer una versión más actual del SQL Server.

Una vez culminada la práctica, se pudo realizar las consultas básicas en la base de datos. Además, con la práctica se pudo comprender mejor el tema de las bases de datos distribuidas y cuáles son sus beneficios y también sus posibles limitaciones. En conclusión, la práctica ha sido de gran ayuda para comprender los temas que anteriormente se habían aprendido conceptualmente.

# **REFERENCIAS**

[1] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

[2] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*. Springer, 2011.

[3] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2007.

[4] T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson, 2014.

[5] C. Ray, *Distributed Database Systems*. CRC Press, 2014.

[6] S. Ceri and G. Pelagatti, *Distributed Databases: Principles and Systems*. McGraw-Hill, 1985.

[7] M. T. Ozsu and P. Valduriez, “Distributed database systems: where are we now?,” *Computer (Long Beach Calif)*, vol. 24, no. 8, pp. 68–78, 1991, doi: 10.1109/2.84879.

[8] W. Litwin, S. Sahri, and T. Schwarz, “Overview of Scalable Distributed Database System SD-SQL Server,” *CERIA, Paris-Dauphine University*, 2021.

[9] V. Narayan, “A SURVEY ON DATA CONSISTENCY AND SECURITY ISSUES IN DISTRIBUTED DATABASE TRANSACTION,” Apr. 2016.

[10] IBM Developer, “Introduction to Client/Server Architecture.” 2021. [Online]. Available: https://developer.ibm.com/technologies/client-server-architecture/

[11] Encyclopaedia Britannica, “Client/Server Architecture.” 2022. [Online]. Available: https://www.britannica.com/topic/client-server-architecture

[12] S. Chandra, S. Kumar, and S. kumar Singh, “An Introduction to Client Server Computing”.

[13] T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education, 2014.

[14] J. F. Buford and H. Yu, “Peer-to-Peer Networking and Applications: Synopsis and Research Directions”, doi: 10.1007/978-0-387-09751-0.

[15] R. Schollmeier, “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications,” *Proceedings - 1st International Conference on Peer-to-Peer Computing, P2P 2001*, pp. 101–102, 2001, doi: 10.1109/P2P.2001.990434.

[16] S. D. Gribble and E. A. Brewer, “Scalable Distributed Data Structures for Internet Service Construction,” *University of California, Berkeley*, 1999, [Online]. Available: https://people.eecs.berkeley.edu/~brewer/cs262b/scalable-ds.pdf

[17] Microsoft, “Partitioning (Database Engine),” *Microsoft Docs*, 2021, [Online]. Available: https://docs.microsoft.com/en-us/sql/relational-databases/partitions/partitioning-overview?view=sql-server-ver15

[18] Oracle, “Partitioning in Oracle Database 12c Release 2,” 2018, [Online]. Available: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/vldbg/partitioning.html

[19] R. Dixon, “Horizontal Partitioning: Scalable SQL Server Database Design,” *Simple Talk*, 2015, [Online]. Available: https://www.red-gate.com/simple-talk/sql/database-administration/horizontal-partitioning-scalable-sql-server-database-design/

[20] A. S. Tanenbaum and M. Van Steen, “Distributed systems: principles and paradigms. Pearson Education.,” p. 633, 2014, Accessed: Apr. 09, 2023. [Online]. Available: www.minix3.org.This

[21] A. González Parra, “Bases de Datos Distribuidas,” 2016.

[22] George. Coulouris, Jean. Dollimore, Tim. Kindberg, J. Belamino. Pulido Junquera, B. Sahelices Fernández, and J. M. Vegas Hernández, “Sistemas distribuidos : conceptos y disen̋o,” 2001.

[23] V. A. López Tulcanaza, “Fragmentación de tablas en base de datos distribuidas. Aplicativo: Implementación de una base da datos distribuida  en el sistema informático de la FICA,” 2011, Accessed: Apr. 09, 2023. [Online]. Available: http://repositorio.utn.edu.ec/handle/123456789/522

[24] P. Bazán, “Aplicaciones, servicios y procesos distribuidos”.

[25] A. S. Tanenbaum and M. van Steen, *Distributed systems: principles and paradigms*. Prentice Hall, 2007.

[26] D. A. Patterson, G. Gibson, and R. H. Katz, “A Case for Redundant Arrays of Inexpensive Disks (RAID)”.

[27] D. Deepa *et al.*, “Distributed Database – SCS1613,” 2022.

[28] S. Ghemawat, H. Gobioff, and S. T. Leung, “The Google file system,” *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 29–43, Oct. 2003, doi: 10.1145/1165389.945450.

[29] S. Mazumdar, D. Seybold, K. Kritikos, and Y. Verginadis, “A survey on data storage and placement methodologies for Cloud-Big Data ecosystem”, doi: 10.1186/s40537-019-0178-3.

[30] V. Toledo and I. Miralles, “Bases de Datos Distribuidas,” 2015, Accessed: Apr. 10, 2023. [Online]. Available: https://iessanvicente.com/colaboraciones/BBDDdistribuidas.pdf

[31] T. Shailesh R, D. LastNameC.A., and G. Ajay B, “Design Distributed Database Strategies for SQMD Architecture,” 2011, Accessed: Apr. 09, 2023. [Online]. Available: http://www.chembiogrid.org/wiki/index.php/Main\_Page

[32] P. O. Orlunwo and A. Prince Oghenekaro, “Distributed database management system (DBMS) architectures and distributed data independence,” 2021, doi: 10.47760/ijcsmc.2021.v10i01.004.

[33] T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education Limited, 2014.

[34] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*. Pearson Education Limited, 2017.

[35] S. Ceri and G. Pelagatti, *Distributed Databases: Principles and Systems*. McGraw-Hill, 1984.

[36] C. Ruiz-Funes and J. E. López, “Niveles de transparencia en sistemas de bases de datos distribuidas,” *Ingeniería, Investigación y Tecnología*, vol. 14, no. 3, pp. 329–342, 2013.

[37] M. Á. Abreu, “Transparencia en Bases de Datos Distribuidas,” *Revista Científica de Computación e Informática*, vol. 5, no. 2, pp. 1–11, 2012.

[38] J. Smith, A. Gounaris, P. Watson, N. W. Paton, and A. A. A. Fernandes, “DISTRIBUTED QUERY PROCESSING ON THE GRID,” 2003, Accessed: Apr. 10, 2023. [Online]. Available: www.ncbi.nlm.nih.gov/BLAST/;

[39] D. Kossmann, “The State of the Art in Distributed Query Processing”, Accessed: Apr. 10, 2023. [Online]. Available: http://www.db.fmi.uni-passau.de/kossmann

[40] K. Beedkar, T. U. Berlin, J.-A. Quiané-Ruiz, and V. Markl, “Compliant Geo-distributed Query Processing,” p. 13, 2021, doi: 10.1145/XXXXXX.XXXXXX.

[41] C. Ray, *Distributed database systems*. Pearson Education India, 2004.

[42] J. A. Hoffer, M. B. Prescott, and H. Topi, *Modern database management*. Pearson, 2015.

[43] A. S. Tanenbaum and M. van Steen, *Distributed systems: principles and paradigms*. Prentice Hall, 2007.

[44] H. Garcia-Molina, J. D. Ullman, and J. Widom, *Database systems: the complete book*. Pearson Education, Inc., 2008.

[45] A. El Abbadi, “Distributed recovery in database systems,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 291–317, 1991.

[46] P. Valduriez, *Distributed database systems*. John Wiley & Sons, 2014.

[47] J. Gray and A. Reuter, “Distributed Commit,” *ACM Transactions on Database Systems (TODS)*, vol. 6, no. 1, pp. 33–53, 1981.

[48] S. B. Navathe and R. Agrawal, “A survey of distributed database recovery protocols,” *ACM Computing Surveys (CSUR)*, vol. 16, no. 4, pp. 439–470, 1984.

[49] H. T. Mouftah, “A survey of fault tolerance in distributed database systems,” *ACM Computing Surveys (CSUR)*, vol. 22, no. 1, pp. 9–32, 1990.

[50] C. Mohan, “What’s new with distributed databases,” *IEEE Computer Society*, vol. 38, no. 6, pp. 20–22, 2005.

[51] J. Gray and A. Reuter, *Transaction processing: concepts and techniques*. Morgan Kaufmann, 1993.

[52] J. Hellerstein, “Parallel and distributed database systems,” *ACM Comput Surv*, vol. 26, no. 2, pp. 132–135, 1994.

[53] P. Valduriez, “Distributed databases,” *ACM Comput Surv*, vol. 28, no. 2, pp. 174–175, 1996.

[54] A. El Abbadi, “Distributed databases,” *ACM Comput Surv*, vol. 22, no. 3, pp. 287–289, 1990.

[55] M. J. Franklin and M. J. Carey, “Distributed database systems,” *ACM Comput Surv*, vol. 22, no. 1, pp. 33–36, 1990.

[56] P. E. Mantey and H. F. Korth, “Distributed database systems: issues and challenges,” *IEEE Computer Society*, vol. 24, no. 12, pp. 6–10, 1991.

[57] S. B. Navathe and A. E. Abbadi, “Distributed databases,” *ACM Comput Surv*, vol. 17, no. 1, pp. 45–74, 1985.