

## **Unidad II. Sistemas de Bases de Datos Distribuidas**

### **M. en C. Euler Hernández Contreras**

#### Contenido

1. Base de datos distribuidas
  - a) Definición
2. Sistemas de clasificación de bases de datos distribuidas
  - a) Clasificación
  - b) Ventajas
3. Procesamiento distribuido y bases de datos distribuidas
4. Características de transparencia de bases de datos distribuidas
5. Fragmentación de los datos
6. Replicación de los datos
7. Colocación de los datos

#### Referencia Bibliográfica

1. Ramez Elmasri, Shamkant B. Navathe. Fundamentos de Sistemas de Bases de Datos, Quinta Edición. Pearson/Addison Wesley, Madrid España 2007, págs. 988
2. Peter Rob, Carlos Coronel. Sistemas de Bases de Datos, Quinta Edición. Thompson Learning Course Technology, México DF 2004, págs. 838
3. Jeffrey A. Hoffer, Mary B. Presscott, Heikki Topi. Modern Database Management, Ninth Edition. Pearson/Prentice, Estados Unidos. 2009
4. [https://msdn.microsoft.com/es-es/library/ms152531\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ms152531(v=sql.120).aspx)

## Bases de Datos Distribuidas

### Definición

“Es una colección lógicamente interrelacionada de datos compartidos (junto con una descripción de estos datos) físicamente distribuidos por una red informática”

“Es una simple base de datos lógica que está distribuida físicamente en diversas computadoras, las cuales están interconectadas con un enlace de comunicación de datos”

### SGBD Distribuido:

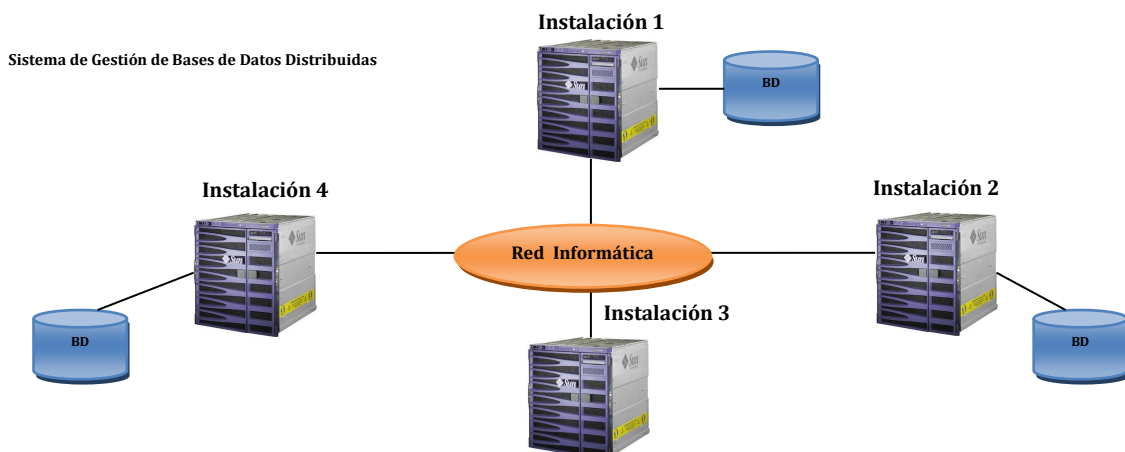
“El sistema software que permite gestionar la base de datos distribuida y hace que dicha distribución sea transparente para los usuarios”

Un *sistema de gestión de bases de datos distribuidas* (SGBDD) está compuesto por una única base de datos lógica dividida en una serie de fragmentos. Cada fragmento se almacena en una o más computadoras bajo el control de un SGBD independiente, estando dichas computadoras conectadas mediante una red de comunicaciones. Cada una de las instalaciones es capaz de procesar de forma independiente las solicitudes de los usuarios que requieran acceso a los datos locales y también es capaz de procesar los datos almacenados en otras computadoras de la red.

Para que un SGBDD pueda ser considerado como tal, deberá disponer al menos de una aplicación global. Un SGBDD tiene, por tanto, las siguientes características.

- Una colección de datos compartidos lógicamente relacionados.
- Los datos están divididos en una serie de fragmentos
- Los fragmentos pueden ser replicados
- Los fragmentos/réplicas se asignan a distintas instalaciones
- Los datos de cada instalación están bajo control de un SGBD
- El SGBD de cada instalación puede gestionar las aplicaciones locales de manera autónoma
- Cada SGBD participa en al menos una aplicación global.

No es necesario que todas las instalaciones del sistema tengan su propia base de datos local, como se muestra en la siguiente Figura.



## Sistemas de Bases de datos Distribuidas [3]

### a) Clasificación

La habilidad para crear una base de datos distribuida ha existido desde los 80's., como ha de esperarse una variedad de opciones de bases de datos distribuidas existen. En la Figura 1 se muestra un rango de ambientes de bases de datos distribuidas. Estos ambientes son brevemente explicados.

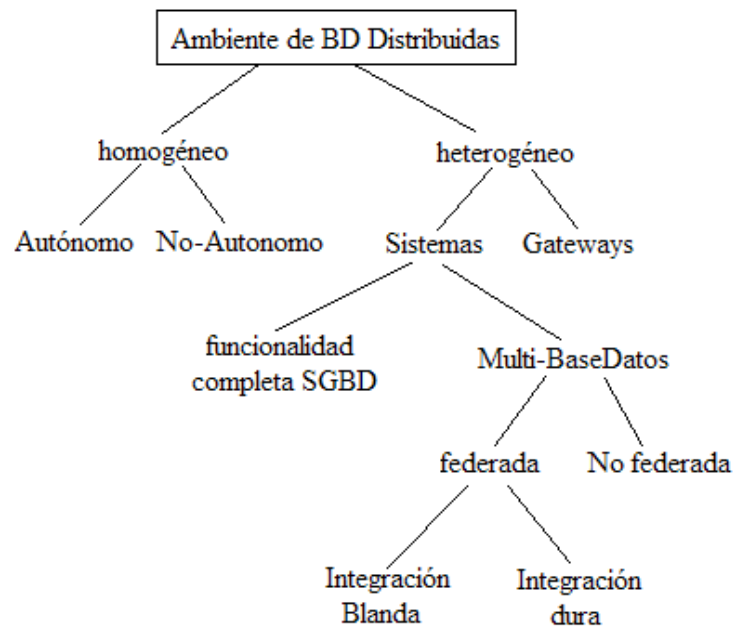


Figura 1. Ambiente de base de datos distribuidas

1. **Homogeneo.** El mismo SGBD es usado en cada nodo.

a) *Autónomo.* Cada SGBD trabaja independientemente pasando mensajes de un lugar a otro para compartir datos actualizados.

b) *No-autónomo.* Existe un SGBD central o maestro que coordina el acceso a la base de datos y la actualización a través de los nodos.

2. **Heterogéneo.** Diferentes SGBD son usados en cada nodo.

2.1 *Sistemas.* Soportan algunas o todas las funcionalidades de una bases de datos lógica.

2.1.1 *Funcionalidad completa de un SGBD.* Soporta toda la funcionalidad de una base de datos distribuidas.

2.1.2 *Multi base datos.* Soporta algunas características de una base de datos distribuidas.

2.1.2.1) *Federada*. Soporta bases de datos locales para requerimientos de datos únicos

a) *Integración blanda*. Existencia de muchos esquemas por cada base de datos local y cada SGBD locales deberían comunicarse con todos los esquemas locales.

b) *Integración dura*. Existe un esquema global que define todos los datos a través de todas las bases de datos locales.

2.1.2.2) *No federadas*. Requiere de todos los accesos para llegar a un módulo de coordinación central.

2.2 Gateways. Simples vías son creadas para otras bases de datos sin el beneficio de una base de dato lógica.

#### b) Ventajas de las bases de datos distribuidas

La administración de bases de datos distribuidas ha sido propuesta por varias razones que van desde la descentralización organizativa y el procesamiento a una mayor autonomía. A continuación haremos referencia a estas ventajas [1]:

1. Administración de datos distribuidos con distintos niveles de transparencia.

Un SGDB debe ocultar los detalles de dónde está físicamente ubicado cada archivo (tabla, relación) dentro de un sistema.

#### Tipos de Transparencia

| Transparencia                          | Descripción   |
|--|---|
| Transparencia de red o de distribución | Hace referencia a la autonomía del usuario de los detalles operacionales de la red.   |
| Transparencia de localización          | Hace mención de que el comando usado para llevar a cabo una tarea es independiente de la ubicación de los datos y del sistema donde se ejecuto dicho comando.                             |
| Transparencia de denominación          | Implica que una vez especificado un nombre, puede accederse a los objetos nombrados sin ambigüedad y sin necesidad de ninguna especificación adicional.                                   |
| Transparencia de replicación           | Almacenarse copias de los datos en distintos lugares para disponer de una mayor disponibilidad, rendimiento y fiabilidad. Permite que el usuario no se entere de la existencia de copias. |
| Transparencia de fragmentación         | Permite que el usuario no se entere de la existencia de fragmentos.   |
| Transparencia de diseño y de ejecución | Hace referencia a la libertad de saber cómo está diseñada la base de datos distribuida y dónde ejecuta una transacción.   |

2. Incremento de la fiabilidad y disponibilidad.

La *fiabilidad* está definida ampliamente como la probabilidad de que un sistema esté funcionando (no caído) en un momento de tiempo, mientras que la *disponibilidad* es la probabilidad de que el sistema esté continuamente disponible en un intervalo de tiempo.

3. Rendimiento mejorado.  
Un SGBD distribuido fragmenta la base de datos manteniendo la información lo más cerca posible del punto donde es más necesaria. La *localización de datos* reduce el enfrentamiento por la CPU y los servicios de E/S, a la vez que atenúa los retardos en el acceso implícito a las redes de área extendida.
4. Expansión más sencilla.  
Es un entorno distribuido, la expansión del sistema en términos de incorporación de más datos, incremento del tamaño de las bases de datos o adición de más procesadores es mucho más sencilla.

### Procesamiento distribuido y bases de datos distribuidas [2].

El *procesamiento distribuido* comparte el procesamiento lógico de la base de datos entre dos o más sitios físicamente independientes conectados mediante una red. Por ejemplo, el procesamiento distribuido podría realizar la entrada/salida (I/O), la selección y la validación de los datos en una computadora, y luego crear un reporte basado en esos datos en otra computadora.

En la Figura 2 se ilustra un ambiente de procesamiento distribuido básico donde se reparte las tareas de procesamiento de la base de datos entre tres sitios conectados mediante una red de comunicaciones. Aunque los datos residen en un solo sitio (México DF), cada sitio puede acceder a los datos y actualizarla. La base de datos se localiza en la computadora A, una computadora de red conocida como servidor de base de datos.

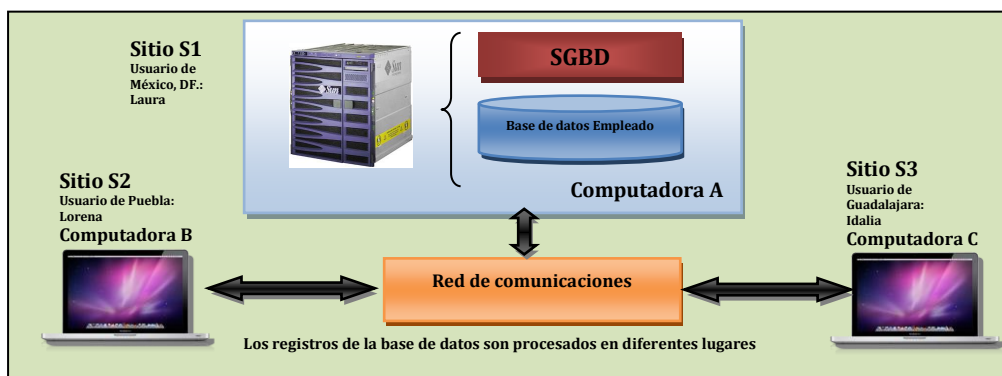


Figura 2. Ambiente de procesamiento Distribuido

Una *base de datos distribuida*, por otra parte, guarda una base de datos lógicamente relacionada en dos o más sitios físicamente independientes. Los sitios están conectados mediante una red de computadoras.

Por su parte, el sistema de procesamiento distribuido utiliza sólo una base de datos en un solo sitio, pero comparte las tareas de procesamiento entre varios. En un sistema de base de datos distribuida una base de datos se compone de varias partes conocidas como fragmentos. Éstos se localizan en diferentes sitios tal como lo muestra la figura en un ambiente de base de datos distribuida.

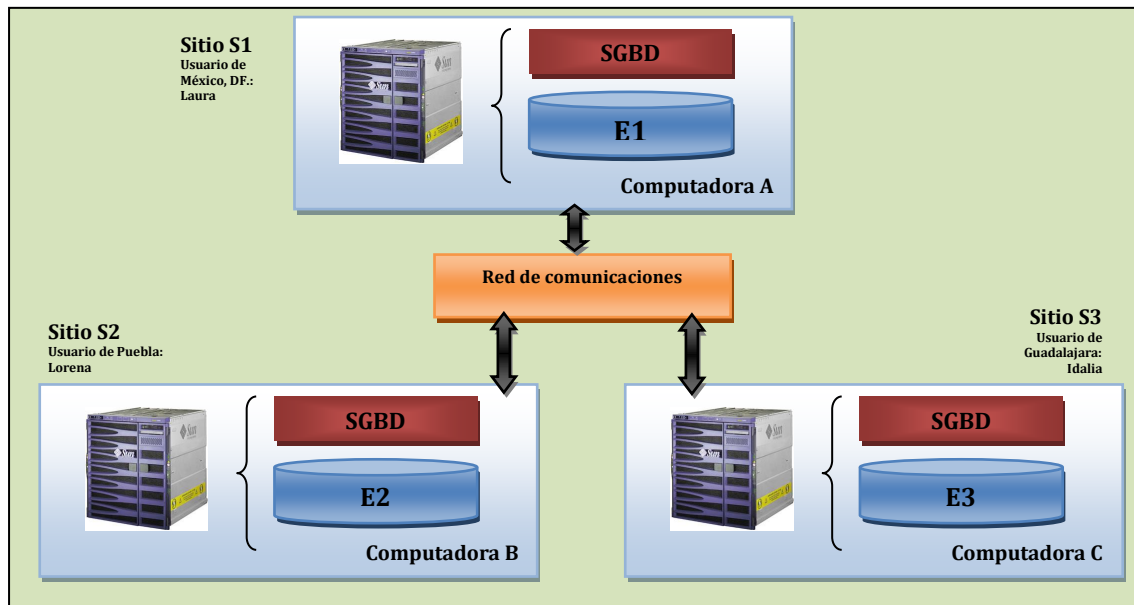


Figura 3. Ambiente de Bases de datos Distribuida

La base de datos de la Figura 3 está dividida en tres fragmentos (E1, E2 y E3) localizados en diferentes sitios. Las computadoras están conectadas por medio de un sistema de red. En una base de datos totalmente distribuida, los usuarios A, B y C, no tienen que saber el nombre o la ubicación de cada uno de los fragmentos para acceder la base de datos. Además, los usuarios de otros sitios diferentes de Monterrey como México DF o Tijuana pueden entrar a la base de datos como una unidad lógica única.

Cuando se examinan y comparan las figuras se deben tomar en cuenta los siguientes puntos:

- A) Procesamiento distribuido no requiere una base de datos distribuida, pero ésta si requiere procesamiento distribuido.
- B) El procesamiento distribuido puede estar basado en una sola base de datos localizada en una sola computadora. Para manejar los datos distribuidos, deben distribuirse copias o partes de las funciones de procesamiento de la base de datos a todos los sitios de almacenamiento de datos.
- C) Tanto el procesamiento distribuido como las bases de datos distribuidas, requieren una red para conectar todos los componentes.

## **Características de transparencia de bases de datos distribuidas**

Estas características tienen la propiedad de permitir que el usuario sienta que es el único que está utilizando la base de datos. En otras palabras, el usuario cree que él o ella están trabajando con un SGBD centralizado.

Las características de transparencia del SGBD son [2]:

- a) **Transparencia de distribución:**  
Permite que una base de datos distribuida sea tratada como una sola base de datos lógica. En un sistema de base de datos distribuida el usuario no necesita saber que los datos están en particiones, que los datos pueden ser replicados en varios sitios y la ubicación de los datos.
- b) **Transparencia de transacción:**  
Permite que una transacción actualice datos en varios sitios de la red. Garantiza que la transacción será o completada en su totalidad o abortada, con lo cual se mantiene la integridad de la base de datos.
- c) **Transparencia de falla:**  
Permite que el sistema continúe operando en el caso de una falla en el nodo.
- d) **Transparencia de desempeño:**  
Permite que el sistema funcione como si fuera un SGBD centralizado. El sistema no sufrirá ninguna degradación de desempeño por su uso en una red o por diferencia de plataforma de la red. Garantiza que el sistema encontrará la ruta de acceso más barata a los datos remotos.
- e) **Transparencia de heterogeneidad:**  
Permite que la integración de varios SGBD locales conforme un esquema común, global. El SGBD Distribuido es el responsable de transformar las solicitudes de datos del esquema global en el esquema de SGBD local.

## **Fragmentación de los datos [2]**

Permite dividir un objeto en dos o más segmentos o fragmentos. El objeto podría ser una base de datos de usuario, una base de datos de sistema o una tabla. Cada fragmento puede guardarse en cualquier sitio en una red de computadoras.

La información de la fragmentación de los datos se guarda en un catálogo de datos distribuidos, desde donde es accedida por el procesador de transacciones para procesar las solicitudes de los usuarios.

Las estrategias de fragmentación están basadas a nivel de tabla y consisten en dividir una tabla en fragmentos lógicos. Tenemos las siguientes estrategias de fragmentación: horizontal, vertical e híbrida.

- a) *Fragmentación horizontal*. Se refiere a la división de una relación en subconjunto (fragmentos) de tuplas. Cada fragmento se guarda en un nodo diferente, y cada uno de ellos tiene filas únicas. Sin embargo, todas las filas únicas tienen los mismos atributos (columnas).
- b) *Fragmentación vertical*. Se refiere a la división de una relación en subconjuntos de atributo (columna). Cada subconjunto (fragmento) se guarda en un nodo diferente, y cada fragmento tiene columnas únicas, con la excepción de la columna clave, la cual es común a todos los fragmentos.
- c) *Fragmentación Híbrida*. Se refiere a una combinación de estrategias horizontales y verticales. En otras palabras, una tabla puede dividirse en varios subconjuntos horizontales (filas), y cada una tiene un subconjunto de los atributos (columnas).

### Replicación de Datos [2][4]

La *replicación* es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La replicación permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

La replicación de datos se refiere al almacenamiento de copias de datos en sitios múltiples servidos por una red de computadoras. Pueden guardarse copias de fragmentos en varios sitios para satisfacer requerimientos de información específicos. Como la existencia de copias de fragmentos puede mejorar la disponibilidad de los datos y el tiempo de respuesta, estas copias reducen los costos de comunicación y de consulta totales.

Suponga que la base de datos A está dividida en dos fragmentos: A1 y A2. Dentro de una base de datos distribuida replicada, es posible el escenario ilustrado en la Figura 4: el fragmento A1 se guarda en los sitios S1 y S2, mientras que el A2 se guarda en los sitios S2 y S3.

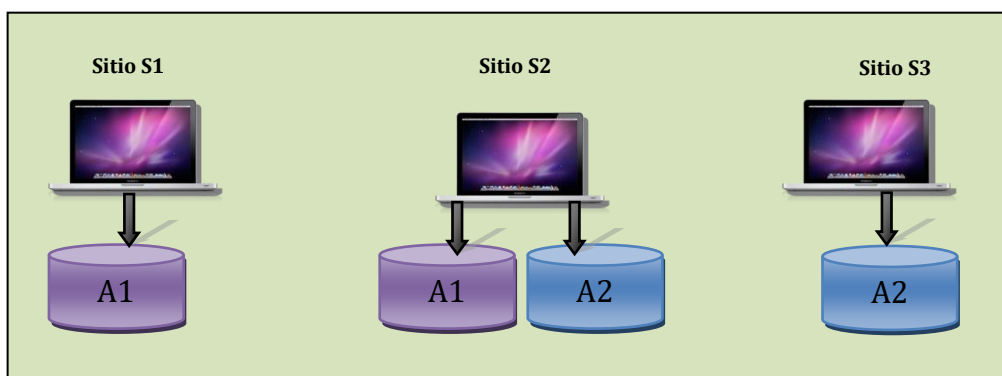


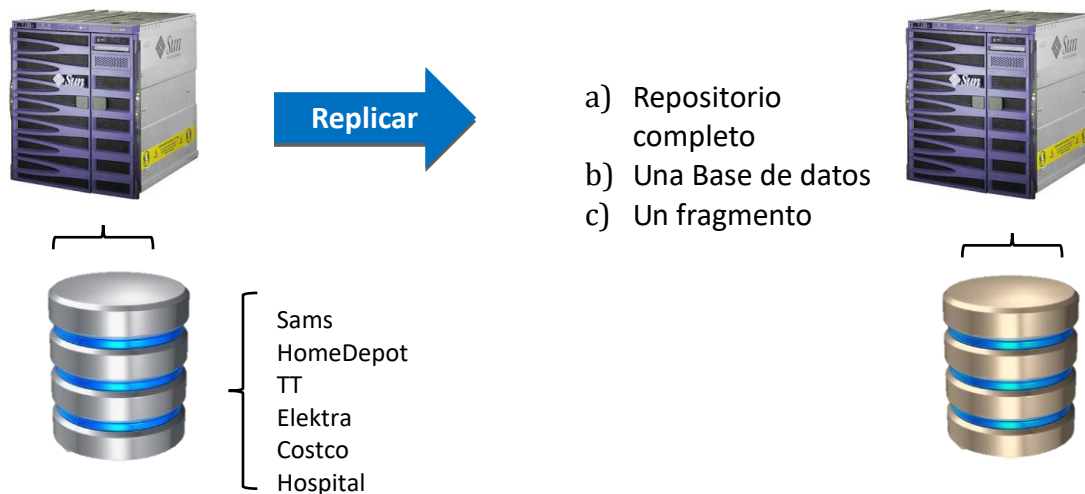
Figura 4. Replicación de Datos



Los datos replicados se someten a la regla de consistencia mutua. La *regla de consistencia mutua* requiere que todas las copias de fragmentos de datos sean idénticas. Por consiguiente, para mantener la consistencia de los datos entre las réplicas, el SGBD debe garantizar que se realice una actualización de la base de datos en todos los sitios donde existas réplicas.

Aunque la replicación tiene algunos beneficios, también exige más complejidad del procesamiento del SGBD, considere los procesos que el SGBD debe realizar para utilizar la BD:

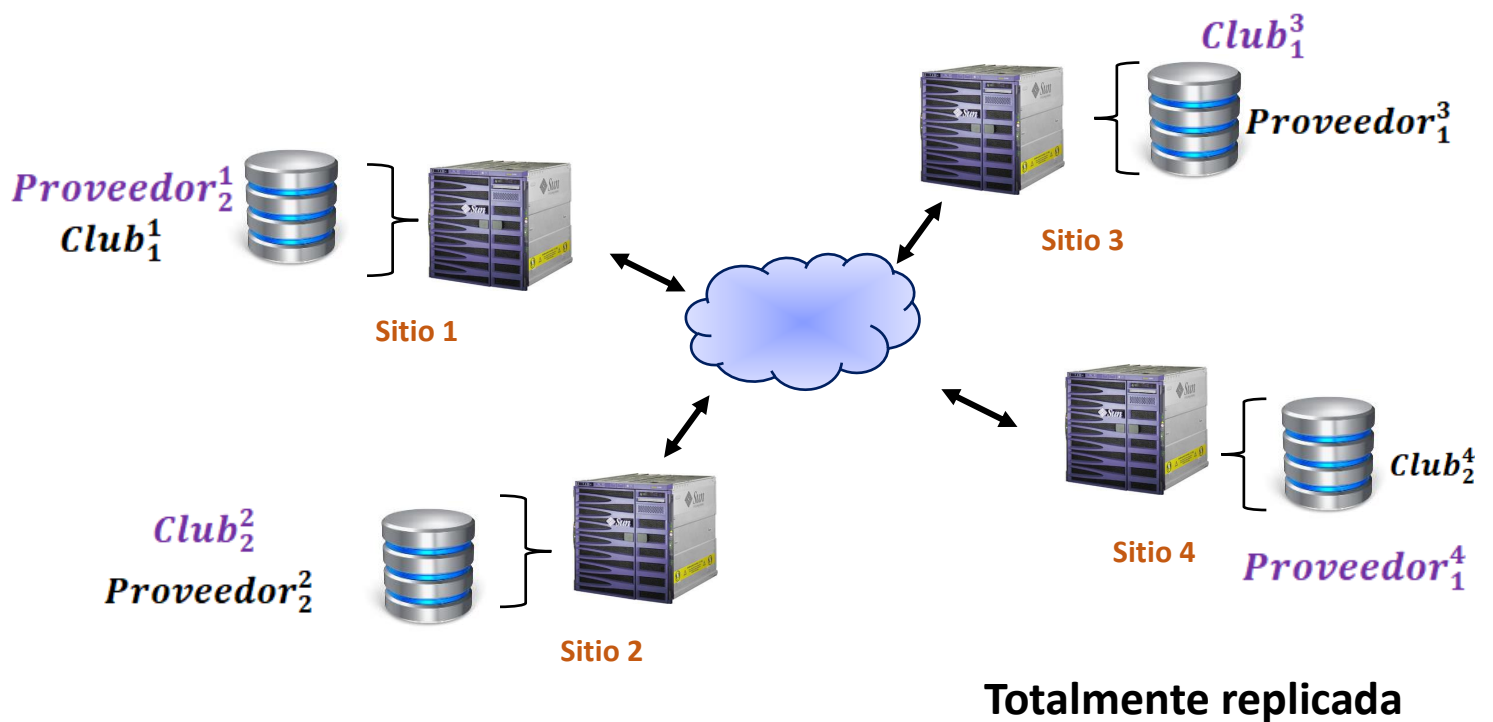
1. Si la base de datos está fragmentada, el SGBD debe decidir qué copia acceder.
2. Una operación READ (lectura) selecciona la copia más cercana para satisfacer la transacción. Una operación WRITE (escritura) requiere que todas las copias se seleccionen y actualicen para satisfacer la regla de consistencia mutua.
3. El procesador de transacciones envía una solicitud de datos a cada procesador de datos para su ejecución.
4. El procesador de datos recibe y ejecuta cada solicitud y envía los datos de vuelta al procesador de transacciones.
5. El procesador de transacciones arma las respuestas del procesador de datos.



Existen tres escenarios de replicación:

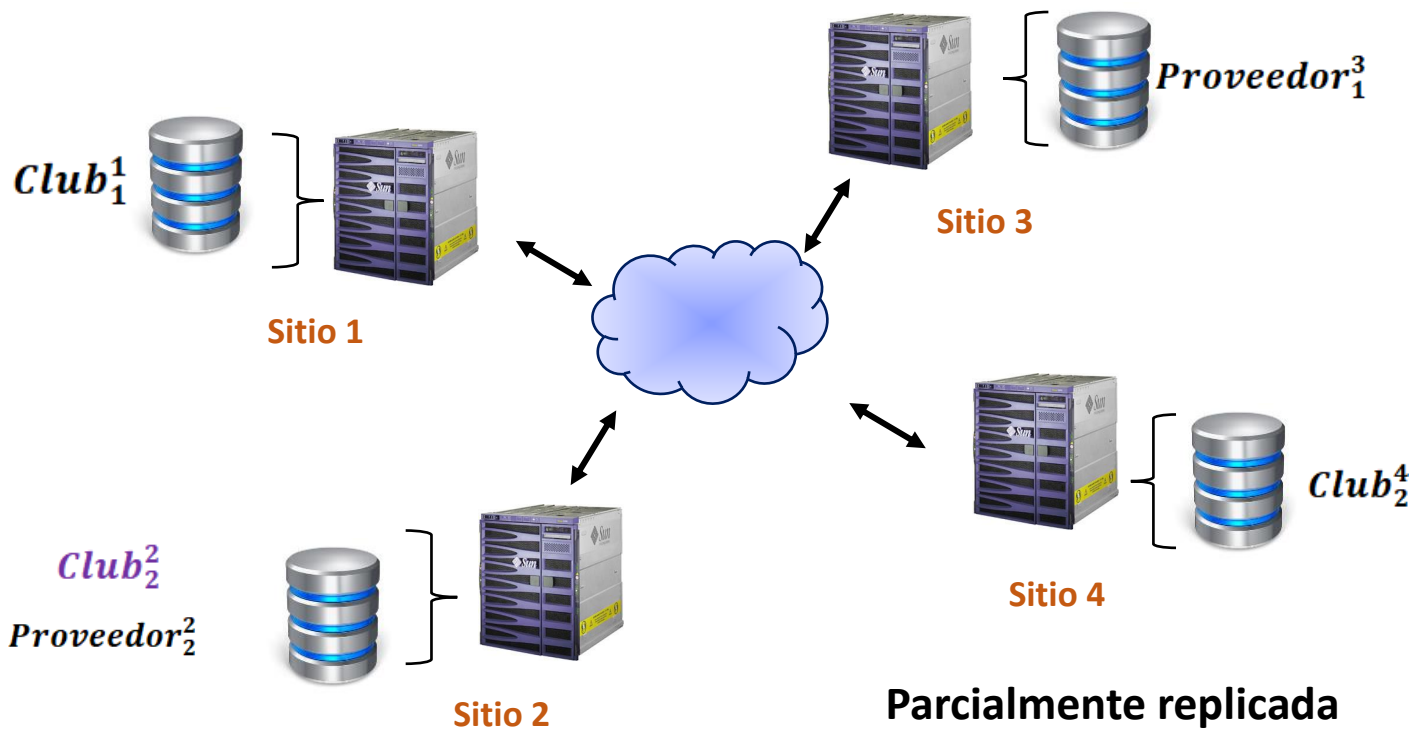
a) Totalmente Replicada.

Guarda varias copias de cada fragmento de la base de datos en varios sitios. En este caso, los fragmentos de la base de datos están replicados. Una base de datos totalmente replicada puede no ser práctica debido la cantidad de carga impuesta al sistema.



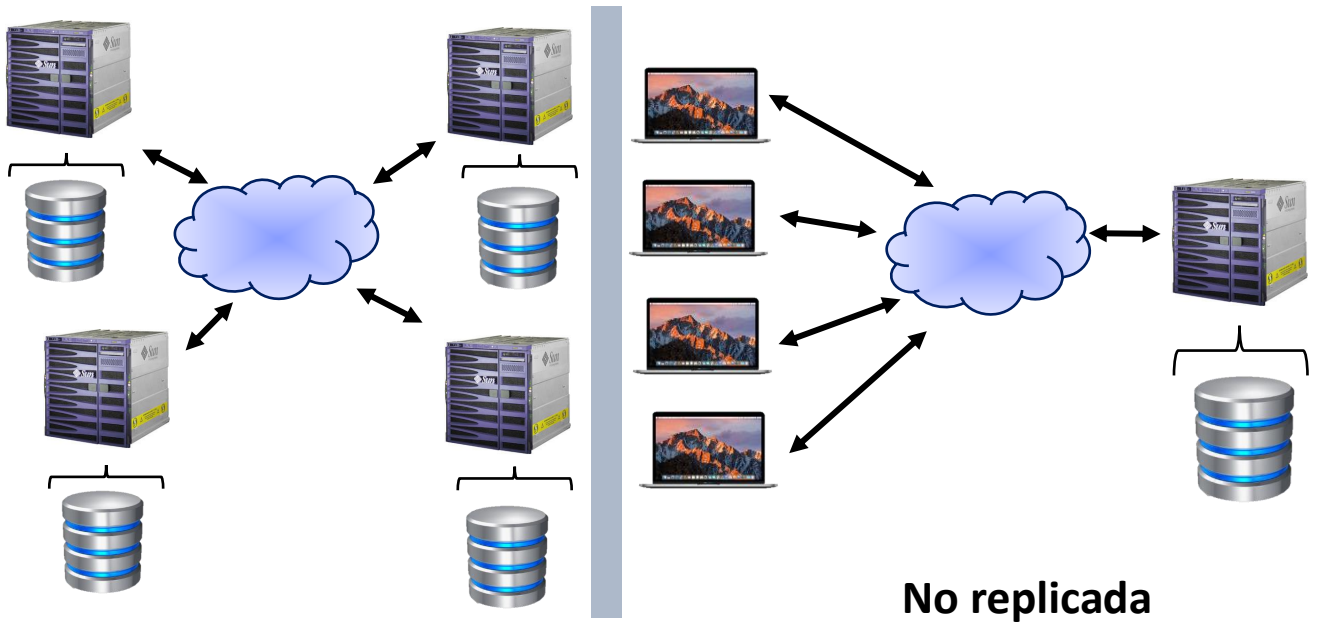
b) Parcialmente Replicada.

Guarda múltiples copias de algunos fragmentos de la base de datos en múltiples sitios. La mayoría de los SGDB son capaces de manejar bien la base de datos parcialmente replicada.



c) No replicada.

Guarda cada fragmento de base de datos en un solo sitio. Por consiguiente, no existen fragmentos de base de datos duplicados.



## Tipos de Replicación

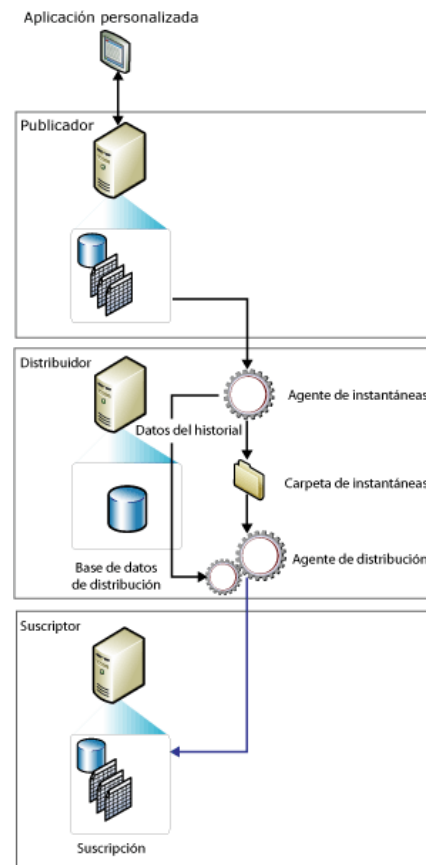
La *replicación transaccional (Transactional Replication)* se usa normalmente en escenarios servidor a servidor que requieren un alto rendimiento, por ejemplo, la mejora de la escalabilidad y la disponibilidad, el almacenamiento de datos y la creación de informes, la integración de datos procedentes de varios sitios, la integración de datos heterogéneos, y la descarga del procesamiento por lotes. La *replicación de mezcla (Merge Replication)* se ha diseñado principalmente para las aplicaciones móviles o de servidores distribuidos que pueden encontrarse con conflictos de datos. Los escenarios más frecuentes son: el intercambio de datos con usuarios móviles, las aplicaciones de punto de venta (POS) a consumidores, y la integración de datos de varios sitios. La *replicación de instantáneas (Snapshot Replication)* se usa para proporcionar el conjunto de datos inicial para la replicación transaccional y de mezcla; también se puede usar cuando está indicada una actualización completa de los datos.

### Replicación de Instantáneas (Snapshot Replication)

El uso independiente de la replicación de instantáneas es más apropiado cuando se cumple una o más de las siguientes condiciones:

- Los datos no cambian con frecuencia.
- Es aceptable disponer de copias de datos desfasados respecto al publicador durante un período de tiempo.
- se duplican pequeñas cantidades de datos.
- Hay un gran volumen de cambios en un corto período de tiempo.

La replicación de instantáneas es más apropiada cuando los cambios de datos son importantes, pero poco frecuentes. Por ejemplo, si una organización de ventas mantiene una lista de precios de productos y todos los precios se actualizan al mismo tiempo una o dos veces al año, es recomendable replicar la instantánea completa de los datos una vez que han cambiado. Para determinados tipos de datos pueden ser adecuadas también instantáneas más frecuentes. Por ejemplo, si una tabla relativamente pequeña se actualiza en el publicador durante el día pero es aceptable cierta latencia, los cambios se pueden entregar por la noche como una instantánea.



### Replicación de Instantáneas

La replicación de instantáneas tiene una carga continua más reducida en el publicador que la replicación transaccional, ya que no se realiza ningún seguimiento de los cambios incrementales. No obstante, si el conjunto de datos que se está replicando es muy grande, será necesaria una cantidad importante de recursos para generar y aplicar la instantánea. Tenga en cuenta el tamaño del

conjunto de datos entero y la frecuencia de los cambios en los datos al evaluar si utiliza o no la replicación de instantáneas.

### Replicación Transaccional (*Transactional Replication*)

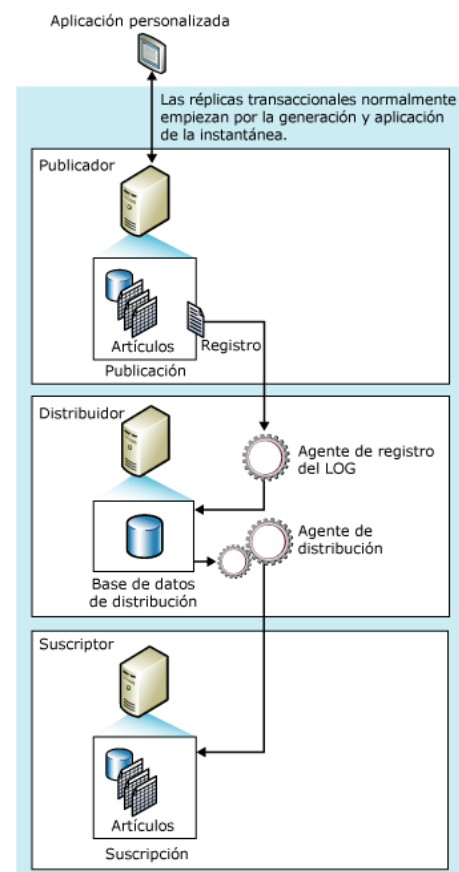
Normalmente, la replicación transaccional se inicia con una instantánea de los datos y los objetos de la base de datos de publicaciones. En cuanto se obtiene la instantánea inicial, los posteriores cambios de datos y modificaciones del esquema realizados en el publicador habitualmente se entregan en el suscriptor cuando se producen (casi en tiempo real). Los cambios de datos se aplican al suscriptor en el mismo orden y dentro de los mismos límites de la transacción que cuando se produjeron en el publicador. Por tanto, en una publicación, se garantiza la coherencia transaccional.

La replicación transaccional se utiliza normalmente en entornos entre servidores y es la adecuada en los siguientes casos:

- Se desea que se propaguen cambios incrementales a los suscriptores en el momento en que ocurren.
- La aplicación requiere una latencia baja entre el momento en que se realizan los cambios en el publicador y el momento en que llegan los cambios al suscriptor.
- La aplicación necesita acceso a los estados intermedios de los datos. Por ejemplo, si una fila cambia cinco veces, la replicación transaccional permite que una aplicación responda a cada cambio (por ejemplo, la activación de un desencadenador) y no solo al cambio de datos neto en la fila.
- El publicador tiene un volumen elevado de actividad de inserción, actualización y eliminación.
- El publicador o el suscriptor es una base de datos que no es de SQL Server, como Oracle.

De forma predeterminada, los suscriptores de publicaciones transaccionales deben tratarse como de solo lectura, porque los cambios no se propagan de vuelta al publicador. Sin embargo, la replicación transaccional ofrece opciones que permiten realizar actualizaciones en el suscriptor.

La replicación transaccional se implementa con el Agente de instantáneas, el Agente de registro del LOG y el Agente de distribución de SQL Server. El Agente de instantáneas prepara archivos de instantáneas que contienen esquemas y datos de las tablas y objetos de



**Replicación Transaccional**

base de datos publicados, almacena los archivos en la carpeta de instantáneas y registra los trabajos de sincronización en la base de datos de distribución del distribuidor.

El Agente de registro del LOG supervisa el registro de transacciones de cada base de datos configurada para la replicación transaccional y copia las transacciones marcadas para ser replicadas desde el registro de transacciones a la base de datos de distribución, que actúa como una cola de almacenamiento y reenvío confiable. El Agente de distribución copia los archivos de instantáneas iniciales de la carpeta de instantáneas y las transacciones almacenadas en las tablas de la base de datos de distribución a los suscriptores.

Los cambios incrementales realizados en el publicador se transfieren a los suscriptores de acuerdo con la programación del Agente de distribución, que se puede ejecutar continuamente para que la latencia sea mínima o a intervalos programados. Puesto que los datos deben cambiarse en el publicador (cuando se utiliza la replicación transaccional sin las opciones de actualización inmediata ni de actualización en cola), se evita que se produzcan conflictos de actualización. Al final, todos los suscriptores disponen de los mismos valores que el publicador. Si se utilizan las opciones de actualización inmediata o de actualización en cola con la replicación transaccional, las actualizaciones pueden realizarse en el suscriptor y, con la actualización en cola, pueden producirse conflictos.

### **Replicación de Mezcla (*Merge Replication*)**

La replicación de mezcla, como la replicación transaccional, normalmente se inicia con una instantánea de los objetos y datos de una base de datos de publicaciones. Los cambios de datos y las modificaciones de esquema posteriores que se lleven a cabo en el publicador y en los suscriptores se controlan mediante desencadenadores. El suscriptor se sincroniza con el publicador cuando están conectados a la red e intercambian todas las filas que han cambiado entre el publicador y el suscriptor desde la última vez que se produjo la sincronización.

La replicación de mezcla se suele utilizar en entornos de servidor a cliente. La replicación de mezcla es adecuada en las siguientes situaciones:

- Varios suscriptores actualizan los mismos datos en diferentes ocasiones y propagan los cambios al publicador y a otros suscriptores.
- Los suscriptores necesitan recibir datos, realizar cambios sin conexión y sincronizar más adelante los cambios con el publicador y otros suscriptores.
- Cada suscriptor requiere una partición de datos diferente.
- Se pueden producir conflictos y, cuando ocurren, debe poder detectarlos y resolverlos.
- La aplicación requiere el cambio de datos neto en lugar de acceso a los estados intermedios de los datos. Por ejemplo, si una fila cambia cinco veces en el suscriptor antes de que éste se sincronice con el publicador, la fila cambiará solo una vez en el publicador para reflejar el cambio de datos neto (es decir, el quinto valor).

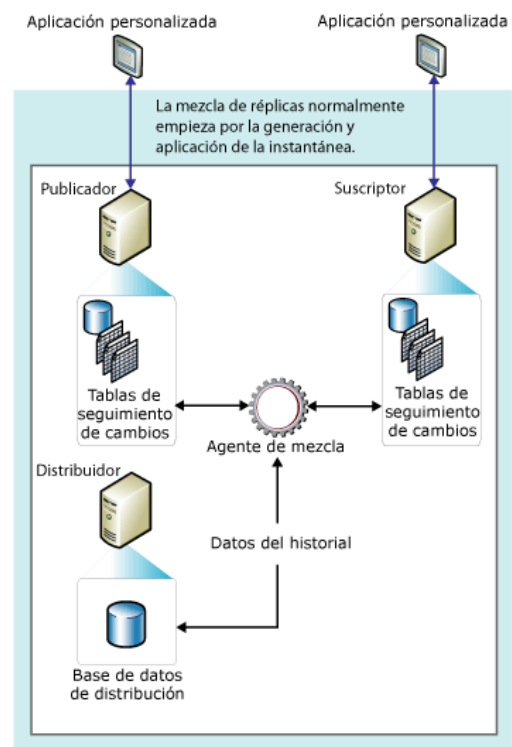
La replicación de mezcla permite que diferentes sitios funcionen de forma autónoma y, después, mezclen las actualizaciones en un solo resultado uniforme. Puesto que las actualizaciones tienen lugar en más de un nodo, los mismos datos pueden haber sido actualizados por el publicador y por más de un suscriptor. Por lo tanto, se pueden producir conflictos al mezclar las actualizaciones y la replicación de mezcla proporciona varias maneras de controlar los conflictos.

El Agente de instantáneas y el Agente de mezcla de SQL Server implementan la replicación de mezcla. Si la publicación no se filtra o utiliza filtros estáticos, el Agente de instantáneas crea una sola instantánea. Si la publicación utiliza filtros con parámetros, el Agente de instantáneas crea una instantánea para cada partición de datos. El Agente de mezcla aplica las instantáneas iniciales a los suscriptores. También combina los cambios incrementales de los datos que tienen lugar en el publicador o en los suscriptores después de la creación de la instantánea inicial y detecta y resuelve los conflictos según las reglas que configure.

Otra taxonomía de replicación:

La *replicación de datos basada en host* usa el software para replicar en tiempo real volúmenes de discos entre clústers ubicados en puntos geográficos distintos. La replicación por duplicación remota permite replicar los datos del volumen maestro del clúster primario en el volumen maestro del clúster secundario separado geográficamente. Un mapa de bits de duplicación remota controla las diferencias entre el volumen maestro del disco primario y el del secundario.

La replicación de datos basada en host es una solución de replicación más económica, porque usa recursos del host en lugar de matrices de almacenamiento especiales.



**Replicación de Mezcla**

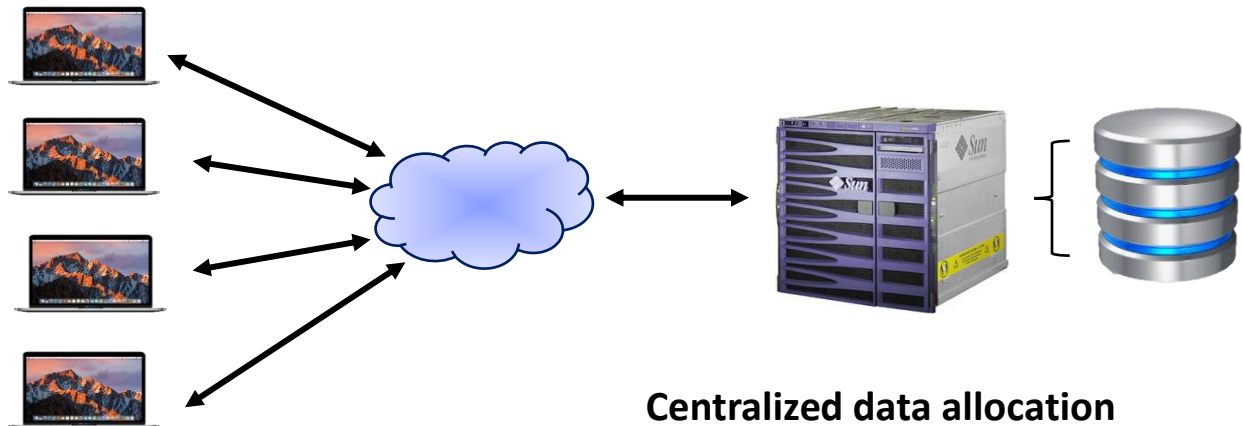
La *replicación de datos basada en almacenamiento* utiliza el software del controlador de almacenamiento para sacar el trabajo de la replicación de datos de los nodos del clúster y moverlo al dispositivo de almacenamiento. Este software libera capacidad de procesamiento del nodo para atender las solicitudes del clúster. La replicación de datos basada en almacenamiento puede ser especialmente importante en configuraciones de clústers de campus y puede simplificar la infraestructura necesaria.



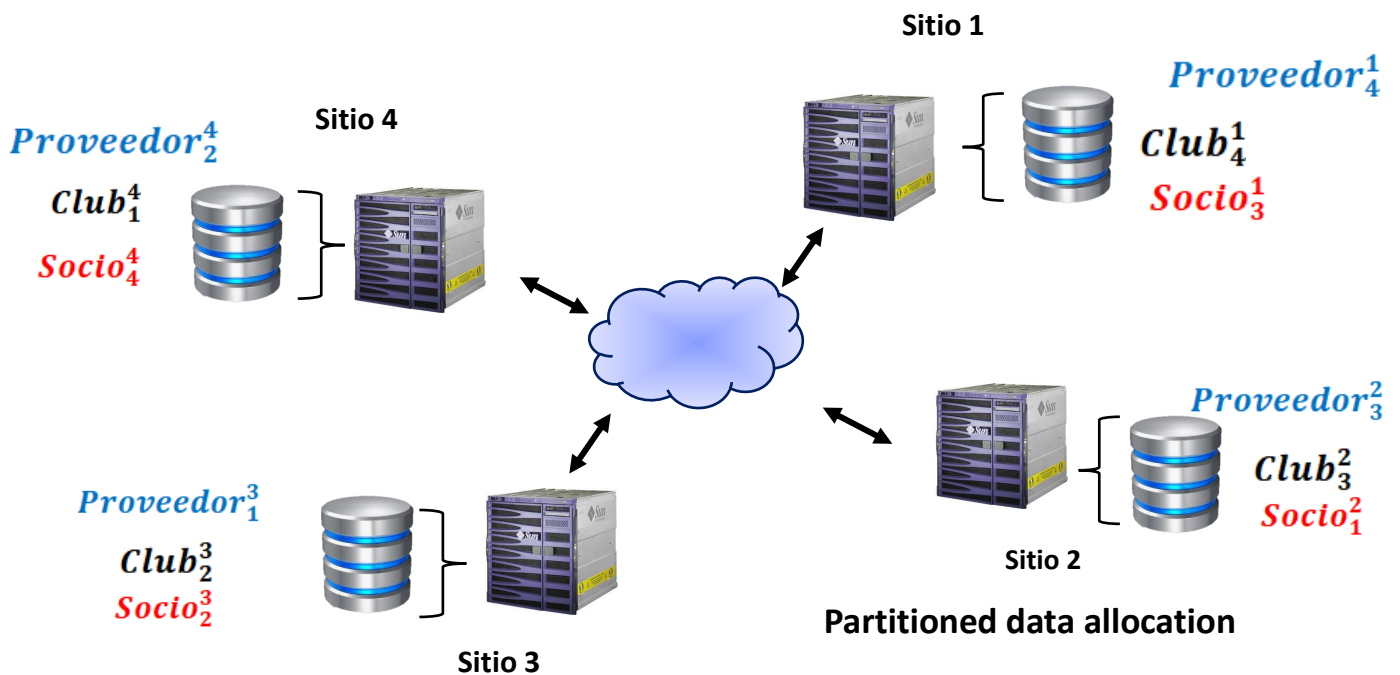
## Colocación de los datos [2]

La colocación de los datos describe el proceso de decidir dónde localizarlos. Las estrategias de colocación de datos son las siguientes:

- a) Colocación centralizada.  
Toda la base de datos es guardada en un solo sitio.

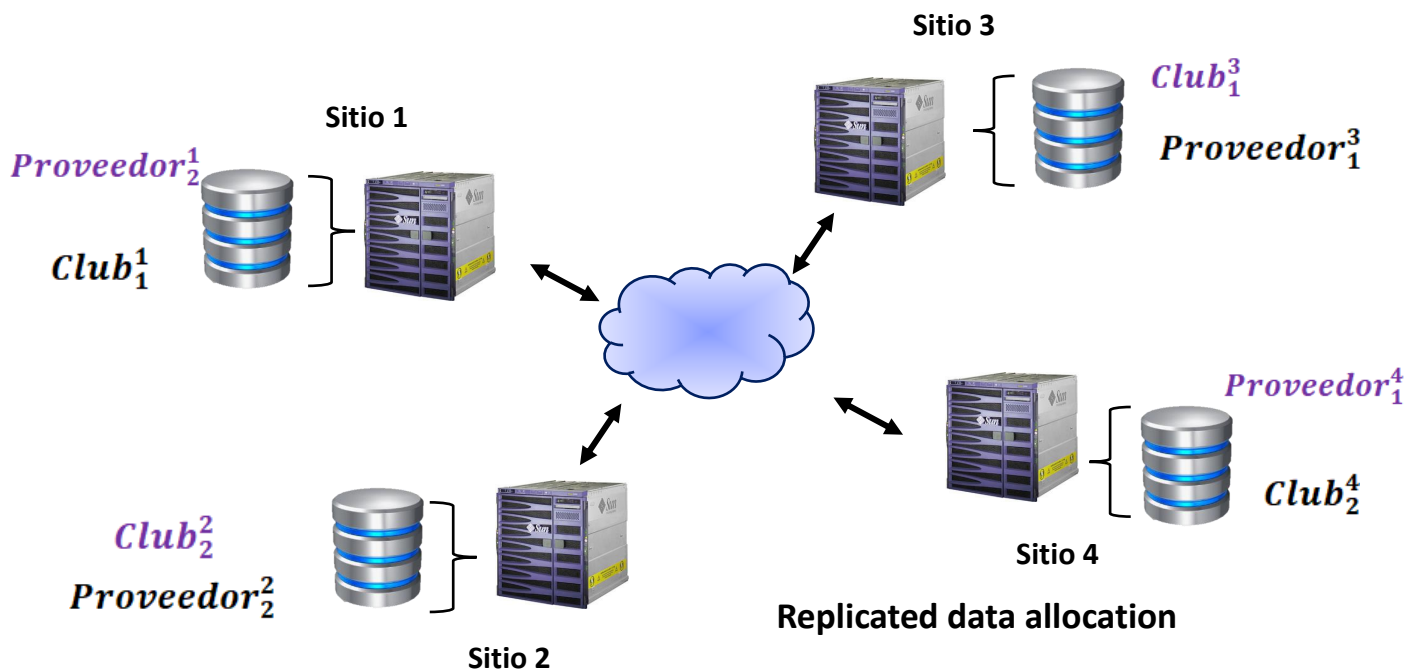


- b) Colocación particionada.  
La base de datos se divide en varias partes desarticuladas (fragmentos) y se guardan en varios sitios.



c) Colocación replicada.

Se guardan copias de uno o más fragmentos de la base de datos en varios sitios.



La colocación de los datos está estrechamente relacionada con la manera en que una base de datos se divide o fragmenta. Los algoritmos de colocación de los datos consideran varios factores, incluidos:

- Objetivos de desempeño y disponibilidad de los datos.
- Tamaño, número de filas y el número de relaciones que una entidad mantienen con otras entidades.
- Tipos de transacciones a ser replicadas a la base de datos, los atributos accedidos por cada una de las transacciones, etc.