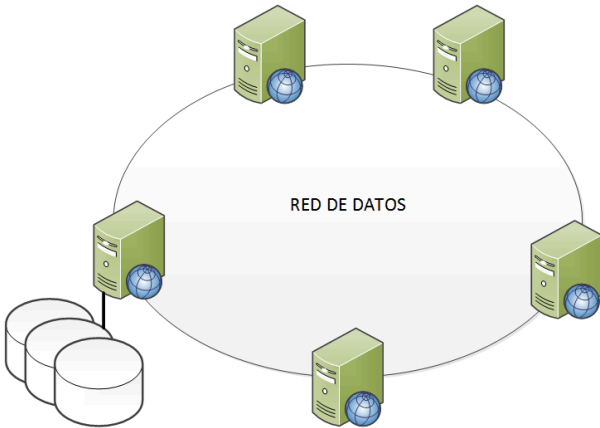
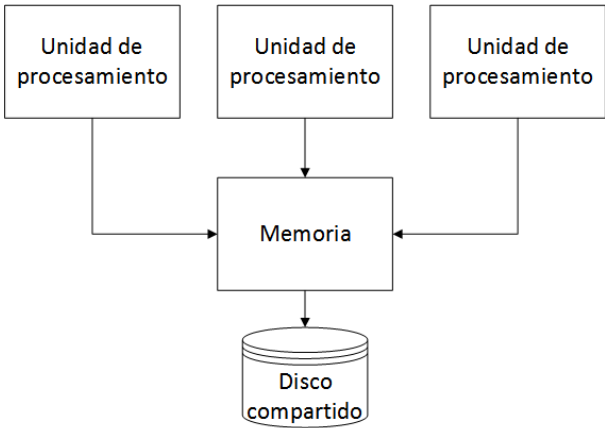
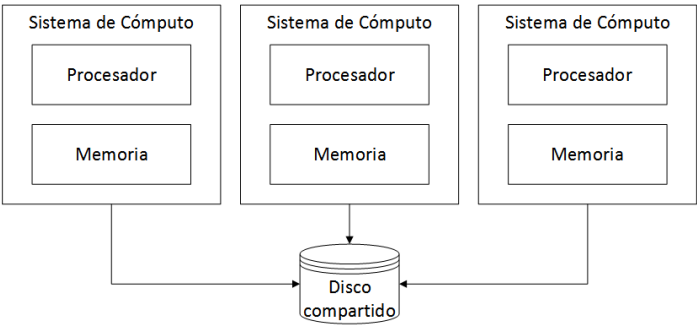
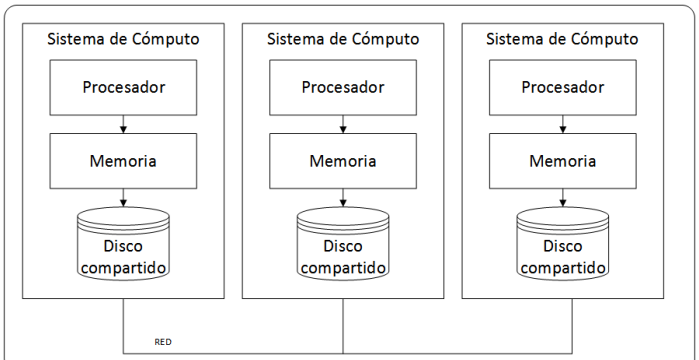


TEMA 1.  
CONCEPTOS BÁSICOS

Material para revisar en clase.

1. ARQUITECTURAS NO CONSIDERADAS COMO UNA BDD.

- Explicar la razón por la cual las siguientes figuras no representan a una BDD.

<p><b>Caso 1:</b> La base de datos se encuentra en un Sitio.</p> 	<p><b>Caso 2:</b> Se tiene un procesador multicore que permite procesamiento paralelo.</p> 
<p><b>Caso 3:</b> Cada sistema de cómputo es un equipo ubicado en un nodo diferente.</p> 	<p><b>Caso 4:</b> En esta arquitectura se tiene un sistema multi – procesador, un procesador por cada nodo. Cada Nodo cuenta con sus propios recursos: memoria, dispositivos de almacenamiento, y periféricos. Todos los nodos son simétricos: idénticas características en cuanto a hardware y software. Una o varias copias del mismo sistema operativo administran a todos los nodos, por ejemplo, el S.O es el responsable de asignar las tareas a cada Nodo. Se comunican a través de una red de datos.</p> 

2. RELACIÓN ENTRE RETOS PARA IMPLEMENTAR UNA BDD

- Las técnicas y decisiones que sea tomadas para resolver los retos al implementar una BDD tienen relación o efecto con otros lo que puede complicar más aun su implementación.
- En las siguientes subsecciones se explica con un mayor detalle las características y acciones que se pueden realizar para implementar cada reto, para al final mostrar la forma en la que una acción o decisión puede tener efectos secundarios sobre otra funcionalidad. Al final se presenta un ejercicio para entender estas dependencias/relaciones.

**Diseño de la base de datos distribuida.**

¿De qué forma será distribuida la base de datos en los diferentes nodos?

- Particionada: la BD es dividida en un cierto número de particiones disjuntas, cada una almacenada en un sitio diferente.
- Replicada:
  - Parcialmente replicada*: cada fragmento es almacenada en más de un sitio, pero no en todos.

- *Totalmente replicada*: la base de datos completa se guarda en todos los sitios
- La combinación de estos conceptos debe permitir minimizar costos de almacenamiento, procesamiento de consultas y los mecanismos de comunicación.

#### **Administración de un directorio de distribución**

- Contiene toda la información acerca de la distribución de los datos: nombre y ubicación de fragmentos, datos de conexión, etc. Este directorio debe ser conocido y accedido por todos los nodos de la BDD.
- Al igual que en una BD centralizada, este diccionario se debe administrar y mantener.
- ¿cómo administrarlo?
  - De forma distribuida: cada sitio tiene una copia del DDD. Esto implica complejidad para mantener todas las copias actualizadas.
  - De forma centralizada: El DDD se encuentra en un solo sitio. Puede provocar cuellos de botella, ¿qué pasa si el sitio que contiene el BDD se cae?

#### **Procesamiento de consultas distribuidas.**

- Consiste en diseñar los algoritmos necesarios que analicen consultas y sean convertidas en un conjunto de operaciones que serán ejecutadas en cada uno de los sitios.
- ¿Qué estrategia es la más adecuada para minimizar costos?, ¿Cómo saber en qué sitios se deben ejecutar ciertas consultas?

#### **Control de concurrencia distribuido.**

- Consiste en sincronizar el acceso a los datos en cada sitio para garantizar su integridad.
- La condición en la que todos los valores de múltiples copias converjan a un mismo valor se le conoce como consistencia mutua. Garantizando con ello la consistencia de los datos en toda la BDD.
- El control de concurrencia es uno de los aspectos más complicados de implementar. Principales técnicas :
  - Sincronización pesimista
  - Sincronización optimista
  - Ambas estrategias se implementan empleando 2 conceptos: bloqueos y timestamping.

#### **Administración de Deadlocks distribuidos.**

- Esta administración es similar a la que se realiza en los sistemas operativos en donde varios usuarios quieren acceder al mismo recurso: disco duro, etc.
- En BDD un deadlock ocurre particularmente con mecanismos de control de concurrencia basados en bloqueos. Existen diversas técnicas para manejarlos como son: prevención, detección y detección/recuperación.

#### **Confiabilidad y alta disponibilidad.**

- La base de datos debe permanecer disponible y con datos consistentes a pesar de que uno o más sitios estén fuera de línea o no disponibles.
- Una vez que el sitio esté disponible, se debe realizar un proceso de recuperación para que el sitio se actualice con todos los cambios que no se pudieron aplicar mientras estaba fuera de línea.

#### **Replicación.**

Para implementar los esquemas de replicación expuestos anteriormente pueden aplicarse 2 protocolos.

- Eager Replication. Obliga a que todos los cambios deben aplicarse en todas las réplicas de datos antes de que la transacción se considere como completa.
- Lazy replication. Los cambios se aplican en un solo sitio (master), se termina la transacción y después se replican a los demás sitios.

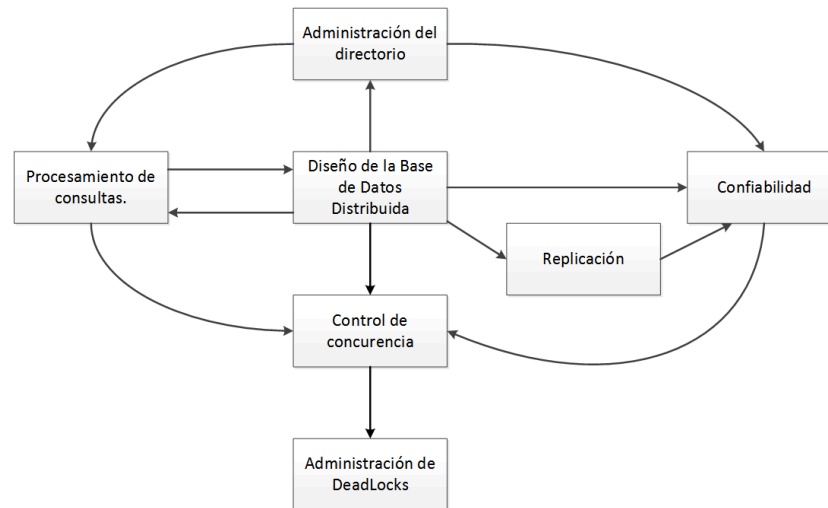
### **3. EJERCICIO: RELACIONES ENTRE RETOS.**

Considerando los siguientes puntos y la siguiente figura, asignar el número correspondiente en los segmentos de recta que conectan a rectángulos y que representan una relación entre 2 retos que son implementados dentro del contexto de una BDD.

- El diseño de una BDD afecta diversas áreas:
  - La definición de fragmentos y su ubicación (diseño de la BDD) determina el contenido del directorio (1).
    - El directorio es empleado por el procesador de consultas para determinar la estrategia de ejecución (2).
      - Los métodos y estrategias de acceso determinadas por el procesador de consultas son empleados como entradas por los algoritmos de distribución y fragmentación de datos (diseño de la BDD) (3).
      - Las decisiones que se realicen en cuanto al diseño de la BDD afectan también el procesamiento de consultas (4).
- La replicación de fragmentos (diseño) afecta a las estrategias de control de concurrencia (5).
- Los métodos de acceso generados por el procesador de consultas también influyen el control de concurrencia (6)

- De no hacerse un correcto control de concurrencia a través del uso de bloqueos, DeadLocks pueden ocurrir. De aquí su estrecha relación (7).
- En cuanto a la confiabilidad, se incluye el diseño de algoritmos de recuperación los cuales afectan los mecanismos de control de concurrencia (8).
  - La manera en la que se distribuye la BDD y se decide cuantas copias existirán (diseño) afecta a dichos mecanismos de recuperación (9). Esta información proviene del diccionario (10).
- Existe una fuerte relación entre la estrategia de replicación de datos y el control de concurrencia debido a que en conjunto deben garantizar la consistencia de datos (11)
- Los protocolos de replicación influyen en las técnicas para garantizar confiabilidad, por ejemplo, los protocolos de commit (12).

Diagrama para el ejercicio.



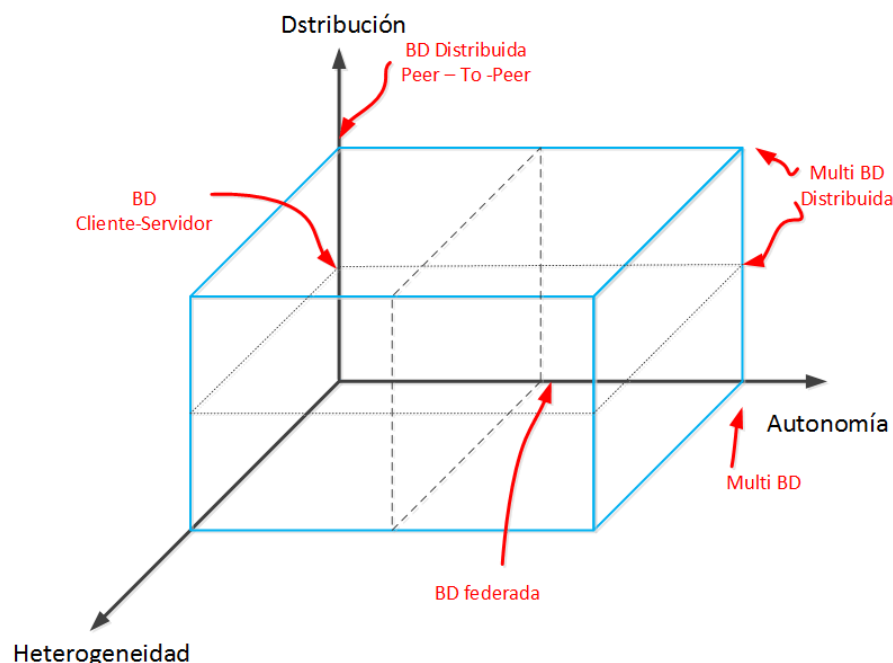
#### 4. EJEMPLO: NIVELES DE TRANSPARENCIA

Suponer que una empresa global decide distribuir los datos de sus clientes y sus adeudos.

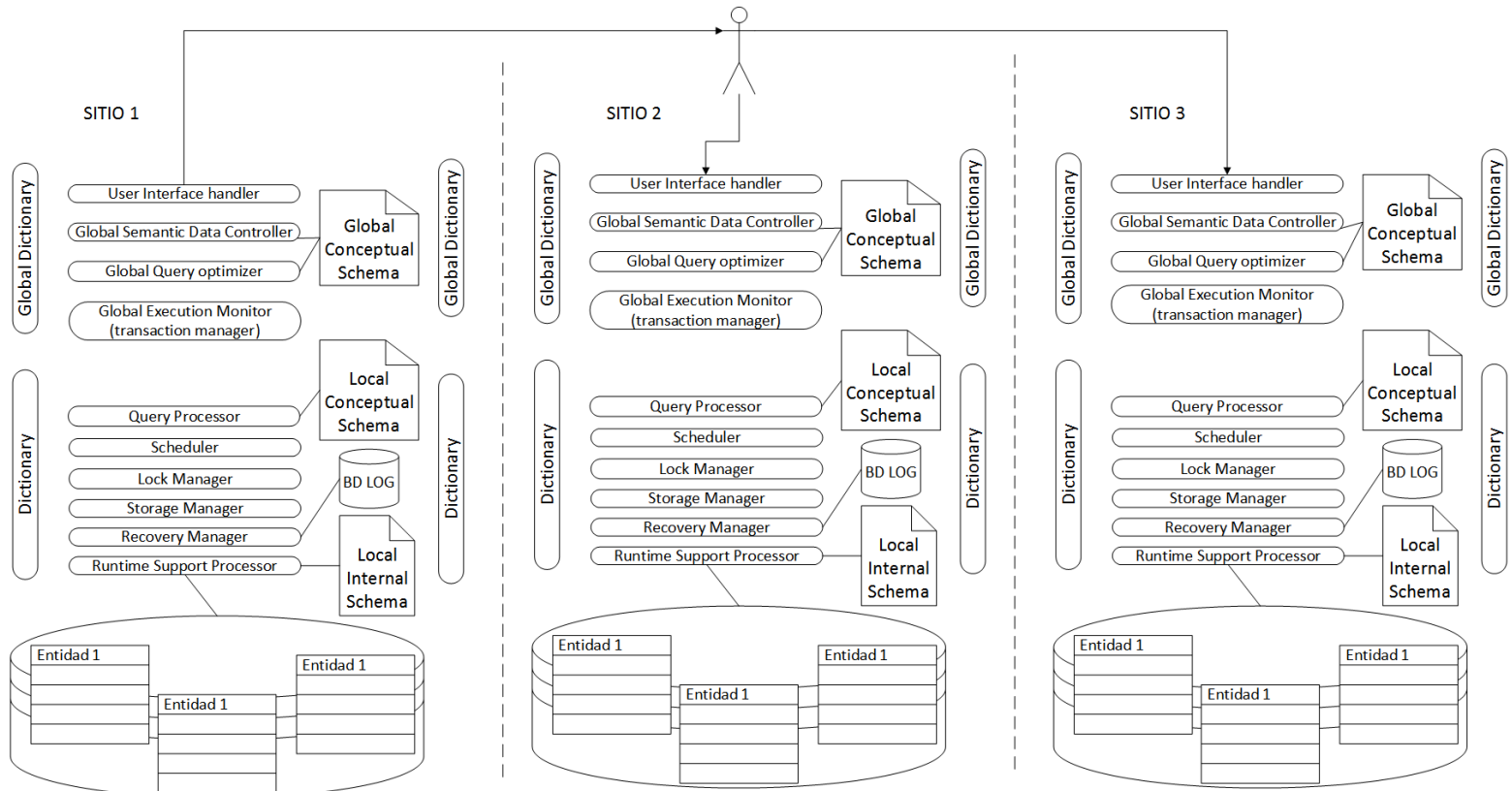
- Se requiere almacenar: nombre, apellidos, adeudo total y país en el que labora. Se consideran las siguientes reglas de fragmentación:
  - Registros de MX en el sitio S1 con hostname `bdmx.com`
  - Registros de JAP en el sitio S2 con hostname `bdjap.com`
- Generar un modelo relacional que represente el esquema global de esta BD.
  - Generar un modelo relacional que represente el esquema de fragmentación
  - Generar 3 consultas que muestre todos los datos de los empleados empleando cada uno de los niveles de transparencia de distribución.

#### 5. ARQUITECTURA DE UN DDBMS

Variables que pueden intervenir en su diseño.



## 6. ARQUITECTURA DE UNA BDD PEER TO PEER



- Por ser un sistema punto a punto, en cada sitio se encuentran los mismos componentes similares a los de una BD centralizada.
- La diferencia es su capacidad extendida para trabajar en un ambiente distribuido. Ejemplos: Lock Manager, Scheduler, Recovery Manager son componentes que existen tanto en BD centralizadas como en BDD.
- La organización física de los datos puede ser diferente en cada sitio. Lo anterior hace necesario la existencia de un Esquema interno local.
- La estructura lógica de los datos se describe en el Esquema conceptual global. Describe la estructura de los datos global en todos los sitios sin considerar la estrategia de fragmentación y replicación, dando la sensación de que la BD no está distribuida, puede visualizarse como la unión de todos los esquemas locales conceptuales.
- Para realizar la correcta administración de la estrategia de fragmentación y replicación se requiere de un Esquema local conceptual. A diferencia del esquema conceptual global, este esquema contiene la definición de las estrategias de fragmentación y replicación en cada sitio (Ver figura anterior).

### Descripción de componentes:

- Administrador de la interfaz de usuario:
  - Se encarga de interpretar las peticiones del usuario, formatear y presentar resultados.
- Controlador de la semántica de los datos:
  - Hace uso de las restricciones (constraints) y las reglas de acceso a datos definidas en el esquema global de la BD para verificar si la consulta puede ser ejecutada con base al esquema de seguridad configurado.
- Optimizador global de consultas:
  - Determina la estrategia o plan de ejecución de la consulta para minimizar costos.
  - Transforma la consulta global en un conjunto de consultas locales haciendo uso de los esquemas locales de cada sitio.
  - Entre otras cosas determina la mejor estrategia para realizar los joins requeridos para ejecutar la consulta.
- Monitor de ejecución distribuido:
  - Coordina la ejecución de la sentencia. A este componente también se le conoce como administrador de transacciones.
- Procesador de consultas:
  - En conjunto con los siguientes componentes, realiza la ejecución de una consulta: Scheduler, Lock Manager, Storage Manager.
- Administrador local de recuperación:
  - Su tarea es asegurar que la bd local permanece consistente, aunque ocurran fallas.
- Procesador de soporte en tiempo de ejecución.
  - Realiza el acceso físico a los datos con base a la salida generada por el optimizador.
  - Representa la interface con el sistema operativo.
  - Contiene el buffer de datos llamado también caché manager.