



Informe Técnico: Arquitecturas de Bases de Datos

ASIGNATURA: BASE DE DATOS II

APELLIDOS Y NOMBRES : CONDOR HUAMAN ERICK JHOSHUA

DOCENTE : MG . RAUL BEJARANO FERNANDEZ

HYO – 2025

Informe Técnico: Arquitecturas de Bases de Datos

1. Introducción

Este informe tiene como objetivo comparar tres tipos de arquitecturas de base de datos: Centralizada, Cliente-Servidor y Distribuida. Se analizan sus definiciones y características, ventajas y desventajas en distintos entornos tecnológicos, ejemplos de implementación en DBMS actuales, y se hace una recomendación de cuál sería la arquitectura más adecuada para un sistema de gestión académica.

2. Definición y características de cada arquitectura

Arquitectura	Definición	Características principales
Centralizada	Una base de datos centralizada es aquella en la que todos los datos se almacenan, mantienen y gestionan en un único servidor o ubicación física.	<ul style="list-style-type: none">• Un único punto de almacenamiento de datos.• Administración, backup y controles centralizados.• Menor complejidad en diseño inicial.• Dependencia fuerte del servidor central.• Requiere buena conexión de red.
Cliente-Servidor	En esta arquitectura, los clientes se conectan a uno o varios servidores que gestionan la base de datos, atienden las peticiones y ejecutan consultas.	<ul style="list-style-type: none">• Separación entre cliente y servidor.• Posibilidad de múltiples clientes simultáneos.• Mejor escalabilidad que la centralizada.• Control de permisos y concurrencia.• Posible cuello de botella si el servidor no es suficiente.
Distribuida	En una arquitectura distribuida, los datos se distribuyen en múltiples nodos / servidores, posiblemente en ubicaciones geográficas diferentes.	<ul style="list-style-type: none">• Diversos nodos que pueden servir datos y procesar peticiones.• Mayor tolerancia a fallos.• Mejor rendimiento para usuarios dispersos geográficamente.• Complejidad en coordinación, consistencia y replicación.• Costos mayores.

3. Ventajas y desventajas en distintos entornos tecnológicos

Se presenta el siguiente cuadro comparativo con las ventajas y desventajas para cada arquitectura.

Arquitectura	Ventajas	Desventajas
Centralizada	<ul style="list-style-type: none">• Simplicidad, menor tiempo de puesta en marcha.• Administración más fácil: backups, seguridad.• Integridad de datos.• Menores costos iniciales.	<ul style="list-style-type: none">• Punto único de fallo.• Rendimiento decreciente con muchos usuarios.• Problemas de latencia.• Escalabilidad limitada.• Riesgo mayor en seguridad.
Cliente-Servidor	<ul style="list-style-type: none">• Mejor concurrencia que la centralizada.• Permite escalado.• Buen control de seguridad, permisos y roles.• Separación de responsabilidades.• Flexibilidad para mejoras en hardware.	<ul style="list-style-type: none">• Si el servidor tiene problemas, afecta a todos los clientes.• Costos mayores.• Mantenimiento más complicado.• Cuellos de botella.• Latencia para clientes remotos.
Distribuida	<ul style="list-style-type: none">• Alta disponibilidad.• Mejor rendimiento para usuarios dispersos.• Escalabilidad horizontal.• Redundancia y replicación.• Mejor tolerancia de carga.	<ul style="list-style-type: none">• Complejidad de diseño.• Problemas de consistencia.• Costos elevados.• Administración complicada.• Seguridad distribuida.

4. Ejemplos de implementación con DBMS actuales

• Centralizada: SQLite para aplicaciones locales embebidas. **Por ejemplo**, una institución educativa pequeña con un único laboratorio de cómputo desea controlar las inscripciones y pagos de los alumnos. No cuenta con un servidor ni con conexión estable a internet. Por ello, implementan un sistema de escritorio que utiliza SQLite embebido: todos los datos se guardan en un único archivo en la misma PC. Es simple, barato y no necesita administración de servidores, aunque si la PC falla, se pierde todo el sistema.

• Cliente-Servidor: MySQL, PostgreSQL, SQL Server. **Por ejemplo**, una universidad mediana con varios campus necesita gestionar matrículas, horarios y calificaciones. Instala un servidor MySQL central en su sede principal. Los administrativos y docentes se conectan como clientes desde sus oficinas a través de la red local o VPN. El servidor central administra la seguridad, las transacciones y los backups. El problema que resuelve es la concurrencia (muchos usuarios accediendo a la vez) y la integridad de la información, pero depende de que ese servidor esté siempre disponible.

- Distribuidas: Apache Cassandra, Apache HBase, Amazon Aurora/RDS, MongoDB. **Por ejemplo**, un consorcio de universidades internacionales comparte expedientes académicos y matrículas de estudiantes en distintos países. Necesitan acceso rápido y tolerancia a fallos incluso si un nodo falla o hay problemas de conexión internacional. Implementan Apache Cassandra con nodos en cada región geográfica para que cada universidad consulte y actualice los datos localmente, con sincronización automática entre nodos. Esto resuelve los problemas de latencia y disponibilidad global, aunque añade complejidad y costos en administración y seguridad.

5. Recomendación para un sistema de gestión académica

Para un sistema de gestión académica que debe estar disponible todo el tiempo, ofrecer seguridad en el manejo de datos sensibles (notas, matrículas, expedientes) y permitir el acceso remoto desde distintas sedes o incluso desde los hogares de docentes y estudiantes, se recomienda adoptar una arquitectura cliente-servidor reforzada con mecanismos de réplica y componentes distribuidos. Esto significa implementar un servidor central de base de datos (por ejemplo, PostgreSQL o MySQL) que actúe como punto principal de escritura y actualización, complementado con réplicas de lectura ubicadas en otras sedes o centros de datos. De esta manera:

- Si el servidor principal falla, una réplica puede asumir temporalmente su rol, evitando caídas del sistema y garantizando alta disponibilidad.
- Se pueden balancear las cargas de trabajo: las consultas masivas (lecturas) pueden dirigirse a las réplicas, mientras las actualizaciones se centralizan en el servidor principal, mejorando así el rendimiento.
- La seguridad se centraliza: se definen políticas de acceso, autenticación, cifrado de datos en tránsito y en reposo desde un punto controlado.
- La escalabilidad es progresiva: se puede empezar con un solo servidor y una réplica, y añadir más nodos a medida que aumentan los usuarios o la cantidad de datos.
- Se minimiza la latencia para usuarios de distintas ubicaciones, ya que pueden conectarse a la réplica más cercana para consultas de lectura.

6. Conclusión y Recomendación Final

La elección de la arquitectura de base de datos es un factor crítico para el éxito y la sostenibilidad de un sistema académico.

Resumen de las Arquitecturas:

- Una arquitectura **Centralizada** es adecuada para instituciones muy pequeñas con pocos usuarios, pero presenta un **punto único de fallo** y limita el rendimiento al crecer.
- Una arquitectura **Totalmente Distribuida** ofrece la máxima disponibilidad y escalabilidad, pero a cambio de una gran **complejidad técnica** y costos muy altos de implementación y administración.

Recomendación Final:

Para una institución educativa con requerimientos de **alta disponibilidad, seguridad** en datos sensibles, **acceso remoto** y necesidad de **crecimiento progresivo**, la solución más equilibrada y práctica es adoptar una arquitectura **Cliente-Servidor con réplicas y componentes distribuidos**.

Este enfoque híbrido aprovecha la centralización para la seguridad y la integridad de las actualizaciones, mientras que las réplicas garantizan la alta disponibilidad y un rendimiento óptimo para las consultas de lectura