**Integrantes del equipo:**

**Sony Gómez: 31.268.878**

**Enrique González: 31.456.812**

**Samuel Molina: 30.922.402**

**Angelo Polgrossi: 31.504.893**

**1. MySQL (Base de Datos Relacional - SQL)**  
MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, basado en el lenguaje SQL (Structured Query Language). Es uno de los más utilizados en aplicaciones web, como WordPress, Joomla y otros sistemas de gestión de contenido.

**Características:**

* **Modelo Relacional:** Organiza los datos en tablas con filas y columnas.
* **Transacciones ACID:** Garantiza la integridad de los datos con propiedades de Atomicidad, Consistencia, Aislamiento y Durabilidad.
* **Escalabilidad Vertical:** Aunque soporta cierta escalabilidad horizontal, su rendimiento se optimiza principalmente aumentando los recursos del servidor.
* **Compatibilidad:** Funciona en múltiples plataformas y es compatible con varios lenguajes de programación como PHP, Java y Python.

**Ventajas:**

* Fácil de usar e implementar.
* Amplia comunidad de soporte.
* Ideal para aplicaciones que requieren integridad y consistencia de datos.

**Desventajas:**

* Limitaciones en entornos de alta concurrencia sin una configuración adecuada.
* Menos flexible para manejar datos no estructurados.

**Casos de Uso:**

* Aplicaciones web y sitios de comercio electrónico.
* Sistemas ERP y CRM.
* Gestión de datos empresariales.

**2. MongoDB (Base de Datos No Relacional - NoSQL)**  
MongoDB es una base de datos NoSQL orientada a documentos. En lugar de tablas, almacena los datos en documentos BSON (una extensión de JSON), lo que permite mayor flexibilidad en el modelado de datos.

**Características:**

**Modelo Basado en Documentos:** Los datos se almacenan en colecciones de documentos JSON.

* **Escalabilidad Horizontal:** Soporta sharding para distribuir los datos en múltiples servidores, mejorando el rendimiento y la disponibilidad.
* **Flexible:** No requiere un esquema fijo, lo que facilita cambios en la estructura de datos.
* **Alta Disponibilidad:** Soporta réplicas para garantizar la redundancia y recuperación ante fallos.

**Ventajas:**

* Ideal para manejar grandes volúmenes de datos no estructurados.
* Rápido en operaciones de lectura y escritura.
* Flexible para adaptarse a cambios rápidos en aplicaciones en desarrollo.

**Desventajas:**

* Menos robusto en transacciones ACID complejas en comparación con bases de datos SQL.
* Mayor consumo de almacenamiento debido a la redundancia de datos.

**Casos de Uso:**

* Aplicaciones en tiempo real, como sistemas de análisis de datos y redes sociales.
* Gestión de contenido, catálogos de productos y aplicaciones móviles.
* Big Data y análisis de grandes volúmenes de información.

**3. Amazon DynamoDB (Base de Datos Escalable)**  
Amazon DynamoDB es un servicio de base de datos NoSQL completamente gestionado, diseñado para aplicaciones de alta disponibilidad y rendimiento. Es altamente escalable, lo que lo hace ideal para entornos con grandes volúmenes de datos y transacciones.

**Características:**

* **Escalabilidad Automática:** DynamoDB ajusta automáticamente la capacidad para manejar picos de tráfico sin intervención manual.
* **Baja Latencia:** Proporciona tiempos de respuesta en milisegundos incluso en grandes cargas de trabajo.
* **Alta Disponibilidad:** Distribuye los datos de forma automática en múltiples regiones para garantizar la redundancia.
* **Transacciones ACID:** Aunque es NoSQL, soporta transacciones ACID para garantizar la integridad de los datos en operaciones críticas.

**Ventajas:**

* Sin necesidad de gestionar la infraestructura, ya que es un servicio completamente administrado por AWS.
* Optimizado para aplicaciones que requieren baja latencia y alta concurrencia.
* Fácil integración con otros servicios de AWS.

**Desventajas:**

* Costos variables según el uso, lo que puede resultar más caro en entornos con alta demanda.
* Menos flexible para consultas complejas en comparación con bases de datos SQL tradicionales.

**Casos de Uso:**

* Aplicaciones móviles, juegos en línea y servicios web con millones de usuarios.
* Comercio electrónico, donde se requiere alta disponibilidad y escalabilidad.
* Sistemas de IoT y análisis en tiempo real.

4) Otras bases de datos: **ACID (Atomicity, Consistency, Isolation, Durability)**

ACID es un acrónimo que representa cuatro propiedades clave que aseguran que las transacciones en una base de datos se procesen de manera confiable. Estas propiedades son esenciales en sistemas donde la integridad de los datos es crítica, como en bancos, sistemas financieros, y aplicaciones empresariales.

**Propiedades de ACID:**

1. **Atomicity (Atomicidad):**
   * Garantiza que una transacción se ejecute completamente o no se ejecute en absoluto.
   * Si ocurre un fallo durante el proceso, la base de datos revierte cualquier cambio parcial, asegurando que los datos no queden en un estado inconsistente.
   * Ejemplo: Si transfieres dinero de una cuenta a otra, ambas operaciones (restar de una cuenta y sumar a la otra) deben completarse juntas.
2. **Consistency (Consistencia):**
   * Asegura que una transacción lleve a la base de datos de un estado válido a otro estado válido, cumpliendo con todas las reglas y restricciones definidas.
   * Si una transacción viola las reglas de integridad, no se aplicará.
   * Ejemplo: No puedes tener un saldo negativo en una cuenta si la lógica del negocio no lo permite.
3. **Isolation (Aislamiento):**
   * Las transacciones concurrentes no deben interferir entre sí.
   * Garantiza que el resultado de las transacciones simultáneas sea el mismo que si se hubieran ejecutado de forma secuencial.
   * Ejemplo: Dos usuarios que actualizan el mismo registro al mismo tiempo no deberían causar conflictos en los datos.
4. **Durability (Durabilidad):**
   * Una vez que una transacción ha sido confirmada (commit), sus cambios son permanentes, incluso si ocurre un fallo del sistema inmediatamente después.
   * Los datos quedan almacenados de forma segura en el disco o en sistemas de respaldo.
   * Ejemplo: Después de realizar un pago en línea, la confirmación del pago permanece registrada, aunque el servidor falle poco después.

**Bases de Datos que Implementan ACID:**

* **Relacionales (SQL):** MySQL, PostgreSQL, Oracle, Microsoft SQL Server.
* **No Relacionales (NoSQL):** Algunas como MongoDB (para operaciones específicas), Couchbase y Amazon DynamoDB (con soporte para transacciones ACID).

**Ventajas de ACID:**

* Garantiza la integridad y confiabilidad de los datos.
* Ideal para aplicaciones críticas donde la precisión de los datos es fundamental.
* Reduce el riesgo de errores en entornos de alta concurrencia.

**Desventajas de ACID:**

* Puede afectar el rendimiento en sistemas de alta carga debido a la necesidad de mantener el aislamiento y la consistencia.
* Menos flexible para aplicaciones que requieren escalabilidad masiva o alta disponibilidad distribuida, donde se prefiere el modelo BASE (para NoSQL).

**Casos de Uso:**

* Sistemas bancarios y financieros.
* Aplicaciones de comercio electrónico que manejan transacciones de pago.
* Sistemas de gestión empresarial (ERP, CRM).