Instituto Nacional de Aprendizaje

Programación de Aplicaciones Informáticas

Modulo: Gestión de Base de Datos

Profesor: Luis Alonso Bogantes.

Estudiante: Mariel Rojas

Cédula: 208030487

*Sistemas Gestores de Base de Datos*

*Fecha: 12/01/2023*

*Sistema Gestor de Base de Datos*

Un sistema gestor de base de datos es el encargado de permitir la administración de las bases de datos (creación de la BD, inserción, modificación, extraer, almacenar, eliminación de datos e información); así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos.

Existen varios tipos de Sistemas Gestores de Base de Datos, como lo es el relacional, el cual se define como un sistema que permite crear un modelo de datos, que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los Sistemas Gestores de Base de Datos Relacional (SQL) establecen relaciones o lazos entre datos y cada una de estas relaciones representan un conjunto de entidades con las mismas propiedades, que están compuestas de una serie de filas o registros llamaos “tuplas”, que dependen de ciertos atributos(columnas).

Los sistemas gestores de base de datos relacionas son una herramienta efectiva que permite varias funciones esenciales tales como:

* + Acceder a los datos a múltiples usuarios al mismo tiempo.
  + Brindar facilidades que garantizan la confidencialidad, la calidad, la seguridad y la integridad de los datos.
  + Control de concurrencia y seguridad.
  + Mecanismos de recuperación.

Existen múltiples sistemas gestores de bases de datos relacionales, entre los más conocidos son:

* SQLite.
* PostgreSQL.
* MS SQL Server.
* MS Access.
* Oracle.
* Sybase.
* MySQL.

Así como existen los Sistemas Gestores de Base de Datos **relacionales**, también existen los sistemas de datos “**No Relacionales**”; Un Sistema Gestor de Bases de Datos No relacional (NoSQL), no requiere de estructuras de datos fijas (tablas). Se hace uso de ellas en entornos distribuidos que deban estar continuamente disponibles y que también administran una gran cantidad de datos.

Los Sistema Gestor de Bases de Datos No relacionales más utilizados actualmente para administrar este tipo de bases de datos son:

* MongoDB.
* Redis.
* Cassandra.

*MongoDB.*

Mongo es un Sistema Gestor de Base de Datos No Relacional por lo que no se basa en tablas y columnas, si no que esta principalmente orientado a documentos, y los datos se almacenan como colecciones o bien como documentos. MongoDB guarda estructuras de datos [BSON](https://es.wikipedia.org/wiki/BSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. (BSON es un formato de intercambio de datos usado principalmente para su almacenamiento y transferencia) .Es una representación [binaria](https://es.wikipedia.org/wiki/Binaria) de [estructuras de datos](https://es.wikipedia.org/wiki/Estructuras_de_datos) y [mapas](https://es.wikipedia.org/w/index.php?title=Vector_asociativo&action=edit&redlink=1). El nombre BSON está basado en el término [JSON](https://es.wikipedia.org/wiki/JSON) y significa 'notación de objeto de JavaScript' Binario.)

Entre este tipo de bases de datos encontramos:

* Clave-valor
* Orientadas a documentos
* Orientadas a columna
* Orientadas a grafo

Las colecciones contienen conjuntos de documentos y funciones, dichas colecciones son el equivalente a las tablas en las bases de datos relacionales clásicas.

. documento sería equivalente a un registro y una colección a una tabla.

En otras palabras, cada base de datos MongoDB contiene colecciones, que a su vez contienen documentos. Cada documento es diferente y puede tener un número variable de campos; El tamaño y el contenido de cada documento también varían.

En la base de datos RDBMS, una tabla puede tener múltiples filas y columnas. De manera similar, en MongoDB, una colección puede tener varios documentos que son equivalentes a las filas. Cada documento tiene múltiples "campos" que son equivalentes a las columnas. Los documentos en una sola colección pueden tener diferentes campos.

El siguiente es un ejemplo de documento basado en JSON.

[Diagrama

Descripción generada automáticamente con confianza media](https://www.tutorialsteacher.com/Content/images/mongodb/document.png)Documento MongoDB

 La implementación en una orientada a documentos podría tener el aspecto representado en la figura siguiente:

Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

La estructura de un documento corresponde a la forma en que los desarrolladores construyen sus clases y objetos en el lenguaje de programación utilizado. En general, las clases no son filas y columnas, sino que tienen una estructura clara formada por pares clave/valor, el cual es un método simple para almacenar en los que una clave sirve como un identificador único, tanto las claves como los valores pueden ser cualquier cosa, desde objetos simples hasta objetos compuestos complejos.

Los documentos no tienen un esquema predefinido y los campos pueden añadirse a voluntad. El modelo de datos disponible en MongoDB facilita la representación de relaciones jerárquicas u otras estructuras complejas.

El desarrollo de MongoDB comenzó en 2007.

2009 MongoDB fue lanzado como un producto independiente y publicado bajo la licencia de código abierto [AGPL](https://es.wikipedia.org/wiki/AGPL).[7](https://es.wikipedia.org/wiki/MongoDB#cite_note-blog.mongodb_1-7)​

En marzo de 2011, se lanzó la versión 1.4 y se consideró ya como una base de datos lista para su uso en producción.[8](https://es.wikipedia.org/wiki/MongoDB#cite_note-8)​

Conclusiones

A diferencia de [las bases de datos SQL](https://datascientest.com/es/sql-vs-nosql-diferencias-usos-ventajas-y-inconvenientes), MongoDB no implica ninguna restricción en cuanto a la estructura de los documentos. Los datos no tienen un esquema preconcebido, y es esta flexibilidad la que hace que MongoDB sea tan potente y eficiente.

MongoDB es un proyecto en constante evolución, por lo que cada versión ofrece nuevas e interesantes mejoras. Como desarrollador, la principal razón para usar MongoDB reside en su velocidad y su capacidad para añadir en una misma colección registros con diferentes campos de manera mucho más flexible que una base de datos basada en un modelo relacional.

Al estar escrito en C++ es multiplataforma e instalarlo en Ubuntu o Windows es relativamente fácil. De hecho, cómo instalar MongoDB en Ubuntu es tan fácil como correr la instrucción sudo apt install -y mongodb desde el terminal, por lo que no hay excusa posible para no darle una oportunidad

Características principales[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=3)]

**Consultas ad hoc**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=4)]

MongoDB soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función definida por el usuario para su mejor ocupación.

**Indexación**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=5)]

Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. El concepto de índices en MongoDB es similar al empleado en [base de datos relacionales](https://es.wikipedia.org/wiki/Base_de_datos_relacional).

**Replicación**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=6)]

MongoDB soporta el tipo de replicación primario-secundario. Cada grupo de primario y sus secundarios se denomina replica set.[13](https://es.wikipedia.org/wiki/MongoDB#cite_note-13)​ El primario puede ejecutar comandos de lectura y escritura. Los secundarios replican los datos del primario y sólo se pueden usar para lectura o para copia de seguridad, pero no se pueden realizar escrituras. Los secundarios tienen la habilidad de poder elegir un nuevo primario en caso de que el primario actual deje de responder.

**Balanceo de carga**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=7)]

MongoDB puede escalar de forma horizontal usando el concepto de *[shard](https://es.wikipedia.org/wiki/Shard_(arquitectura_de_base_de_datos)" \o "Shard (arquitectura de base de datos))*.[14](https://es.wikipedia.org/wiki/MongoDB#cite_note-14)​ El desarrollador elige una clave de sharding, la cual determina cómo serán distribuidos los datos de una colección. Los datos son divididos en rangos (basado en la clave de sharding) y distribuidos a través de múltiples shard. Cada shard puede ser una réplica set. MongoDB tiene la capacidad de ejecutarse en múltiple servidores, balanceando la carga y/o replicando los datos para poder mantener el sistema funcionando en caso de que exista un fallo de hardware. La configuración automática es fácil de implementar bajo [MongoDB](https://tekslate.com/sort-document-mongodb/) y se pueden agregar nuevas servidores a MongoDB con el sistema de base de datos funcionando.

**Almacenamiento de archivos**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=8)]

MongoDB puede ser utilizado como un [sistema de archivos](https://es.wikipedia.org/wiki/Sistema_de_archivos), aprovechando la capacidad de MongoDB para el balanceo de carga y la replicación de datos en múltiples servidores. Esta funcionalidad, llamada GridFS[15](https://es.wikipedia.org/wiki/MongoDB#cite_note-15)​ e incluida en la distribución oficial, implementa sobre los drivers, no sobre el servidor,[16](https://es.wikipedia.org/wiki/MongoDB#cite_note-16)​ una serie de funciones y métodos para manipular archivos y contenido. En un sistema con múltiple servidores, los archivos pueden ser distribuidos y replicados entre los mismos de forma transparente, creando así un sistema eficiente tolerante de fallos y con balanceo de carga.

**Agregación**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=9)]

MongoDB proporciona un framework de agregación que permite realizar operaciones similares al "GROUP BY" de SQL. El framework de agregación está construido como un pipeline en el que los datos van pasando a través de diferentes etapas en los cuales estos datos son modificados, agregados, filtrados y formateados hasta obtener el resultado deseado. Todo este procesado es capaz de utilizar índices si existieran y se produce en memoria. Asimismo, MongoDB proporciona una función MapReduce que puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación.

**Ejecución de JavaScript del lado del servidor**[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=10)]

MongoDB tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas.

Fragmentación (*sharding*)[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=11)]

Si estás desarrollando un servicio que se va haciendo popular o los niveles de acceso a base de datos son cada vez más altos, empezarás a notar que tu base de datos está siendo atacada por un tráfico creciente y tu servidor esté sufriendo por los altos niveles de stress y te podrías ver en la necesidad de actualizar tu infraestructura para soportar la demanda.

Aquí entra en juego el sharding, es el modo en el que hacemos nuestra base de datos escalable. En lugar de tener una colección en una base de datos, la pondríamos en varias bases de datos distribuidas, de modo que a la hora de consultar los datos de dicha colección, los recuperemos como si de una única base de datos se tratase. Mongo se encargará de averiguar de manera transparente en qué base de datos se encuentran los datos.

Los fragmentos estarán formados por replica set, de modo que si creamos tres fragmentos, cada uno de los cuales tiene una replica set con tres servidores, estaríamos hablando de un total de nueve servidores.

Las consultas se realizan de manera distribuida a través de un módulo enrutador, “MongoS”, que mantiene un pool de conexiones a

Aadhar: Aadhar es un proyecto de identificación única de la India y la mayor base de datos biométricos del mundo. MongoDB es una de las bases de datos que utiliza para almacenar los datos biométricos y demográficos de más de 1.200 millones de personas.

eBay: La empresa estadounidense de comercio electrónico eBay, que funciona como B2C y C2C, utiliza MongoDB en sus diversos proyectos, como las sugerencias de búsqueda, la gestión de la nube y el almacenamiento de metadatos.

Shutterfly: Shutterfly es una popular plataforma para compartir fotos que utiliza MongoDB para almacenar y gestionar más de 6.000 millones de imágenes, con una capacidad de transacción de 10.000 operaciones/segundo.

Ventajas

1. MongoDB almacena datos como un documento basado en JSON que no aplica el esquema. Nos permite almacenar datos
2. podríamos concluir que las mayores virtudes de MongoDB están en su capacidad para escalar horizontalmente, en el buen rendimiento que ofrece en general en las operaciones de lectura, en su capacidad para adaptarse a los cambios en la estructura de la información, en las posibilidades que ofrecen sus índices, y en la facilidad que ofrece para programar aplicaciones en muy diversos lenguajes.jerárquicos en un documento. Esto facilita el almacenamiento y la recuperación de datos de manera eficiente.
3. Es fácil de escalar hacia arriba o hacia abajo según el requisito, ya que es una base de datos basada en documentos. MongoDB también nos permite dividir los datos en varios servidores.

## Principales limitaciones[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=12)]

### Implementación de propiedades ACID multidocumento[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=13)]

MongoDB solo garantizaba ACID dentro del mismo documento. El no implementar las propiedades [ACID](https://es.wikipedia.org/wiki/ACID) generaba que la base de datos no aseguraran la durabilidad, la integridad, la consistencia y el aislamiento requeridos obligatoriamente en las transacciones. El soporte para transacciones [ACID](https://es.wikipedia.org/wiki/ACID) de múltiples documentos se agregó a MongoDB en la versión 4.0 de junio de 2018.

Sobre la base de este punto se detallan los cuatro siguientes:

### Problemas de consistencia[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=14)]

En versiones anteriores de la base de datos las lecturas estrictamente consistentes ven versiones obsoletas de documentos, también pueden devolver datos incorrectos de lecturas que nunca deberían haber ocurrido.[17](https://es.wikipedia.org/wiki/MongoDB#cite_note-17)​

Este problema se considera solucionado a partir de la versión 3.4: <https://jepsen.io/analyses/mongodb-3-4-0-rc3>

### Bloqueo a nivel de documento[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=15)]

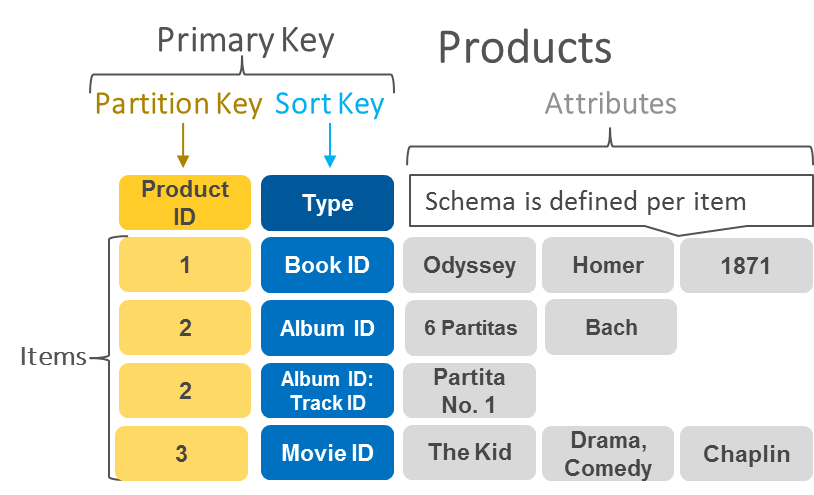
MongoDB bloquea la base de datos a nivel de documento ante cada operación de escritura. Sólo se podrán hacer operaciones de escritura concurrentes entre distintos documentos.

### Las escrituras no son durables ni verificables[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=16)]

MongoDB retorna cuando todavía no se ha escrito la información en el espacio de almacenamiento permanente, lo que puede ocasionar pérdidas de información. En MongoDB 2.2 se cambia el valor por defecto para escribir en al menos una réplica, pero esto sigue sin satisfacer la durabilidad ni la verificabilidad.[18](https://es.wikipedia.org/wiki/MongoDB#cite_note-18)​

### Problemas de escalabilidad[[editar](https://es.wikipedia.org/w/index.php?title=MongoDB&action=edit&section=17)]

Tiene problemas de rendimiento cuando el volumen de datos supera los 100GB.[19](https://es.wikipedia.org/wiki/MongoDB#cite_note-19)​

**

Una de las principales características de MongoDB es que es un sistema es de código abierto. Otra característica importante de MongoDB es la **elasticidad de sus entornos**.

**BSON** es un formato de intercambio de datos usado principalmente para su almacenamiento y transferencia en la base de datos [MongoDB](https://es.wikipedia.org/wiki/MongoDB). Es una representación [binaria](https://es.wikipedia.org/wiki/Binaria) de [estructuras de datos](https://es.wikipedia.org/wiki/Estructuras_de_datos) y [mapas](https://es.wikipedia.org/w/index.php?title=Vector_asociativo&action=edit&redlink=1). El nombre *BSON* está basado en el término [JSON](https://es.wikipedia.org/wiki/JSON) y significa *Binary JSON* (JSON Binario).

**JavaScript Object Notation**, 'notación de objeto de JavaScript')

*Video las transacciones ya fueron agregadas en las ultimas versiones*