

## Descargas

**mongo community server:** <https://www.mongodb.com/try/download/community>  
Compass (Software de management de la db, interfaz gráfica).

**Compass incluye en su interfaz al mongoShell):**

<https://www.mongodb.com/try/download/compass>

**mongoShell** (Es una consola para realizar consultas y/u operaciones que es propia de mongoDB): <https://www.mongodb.com/try/download/shell>

### 1. Bases de Datos No Relacionales (NoSQL)

Las bases de datos no relacionales, o **NoSQL** (Not Only SQL), son sistemas de gestión de bases de datos que permiten almacenar y recuperar grandes volúmenes de datos de una manera **más flexible** que las bases de datos relacionales. Se caracterizan por no seguir el modelo tradicional de tablas y filas, sino que adoptan otros enfoques como **documentos**, grafos, pares clave-valor, o estructuras de columnas.

### 2. Diferencias entre Bases de Datos Relacionales y No Relacionales

Característica	Relacional	No Relacional
<b>Modelo de datos</b>	Basado en tablas y relaciones entre ellas.	Flexible (documentos, clave-valor, etc.)
<b>Estructura</b>	Esquema rígido y definido de antemano (normalización)	Esquema flexible y adaptable
<b>Escalabilidad</b>	Vertical, en general	Horizontal
<b>Consultas</b>	SQL (estructurado y estándar)	Consultas adaptadas según el modelo
<b>Integridad referencial</b>	Soporta integridad con claves foráneas	No suelen tener integridad referencial
<b>Transacciones</b>	ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)	Generalmente BASE (Básico, Disponible, Estado eventual)
<b>Uso ideal</b>	Datos estructurados y relaciones complejas	Datos semi-estructurados o sin estructura

### 3. Pros y Contras

#### Bases de Datos Relacionales

Pros:

- Integridad y Consistencia: Gracias al cumplimiento de las propiedades ACID, garantizan la consistencia de los datos.
- Consultas Complejas: SQL es potente y permite realizar consultas avanzadas.

- Soporte de Relaciones: Ideal para datos con relaciones complejas.

Contras:

- Escalabilidad: La escalabilidad es principalmente vertical, lo que limita su crecimiento cuando los datos aumentan mucho.
- Rigididad del Esquema: El esquema es fijo, lo que implica una planificación cuidadosa antes de su uso.
- Coste de Mantenimiento: La administración de bases de datos relacionales puede ser compleja y costosa.

## **Bases de Datos No Relacionales**

Pros:

- Escalabilidad Horizontal: Pueden escalar añadiendo nodos, ideales para grandes volúmenes de datos.
- Flexibilidad: Los esquemas pueden adaptarse y cambiar sin problemas.
- Velocidad: Óptimas para operaciones de lectura y escritura rápidas, especialmente en datos sin relaciones complejas.

Contras:

- Falta de Integridad Referencial: No suelen tener soporte para relaciones estrictas entre datos.
- Consistencia Eventual: Algunos sistemas NoSQL sacrifican la consistencia por la disponibilidad y velocidad.
- Consultas Limitadas: Las consultas complejas pueden ser más difíciles de implementar, dependiendo del modelo NoSQL.

## **4. MongoDB: Características, Ventajas, Historia e Importancia**

### Características Principales de MongoDB

MongoDB es una base de datos NoSQL orientada a documentos, en la que los datos se almacenan en estructuras BSON (similar a JSON). Es muy popular en aplicaciones modernas debido a su flexibilidad y eficiencia en manejar datos no estructurados o semi-estructurados.

- Modelo de Documento: Los datos se organizan en documentos BSON, lo que permite un formato similar a JSON.
- Esquema Flexible: No requiere esquemas fijos, lo que facilita adaptarse a cambios en los datos.
- Escalabilidad Horizontal: Diseñada para escalar distribuyendo datos en varios servidores.
- Índices y Agregaciones: Soporta índices y permite operaciones de agregación avanzadas para consultas eficientes.
- Compatibilidad con Replicación y Alta Disponibilidad: Ofrece replicación automática y distribución de datos para asegurar la disponibilidad.

## Ventajas de MongoDB

- Desarrollo Rápido: Esquemas flexibles permiten ajustar los datos fácilmente.
- Distribución Geográfica de Datos: Perfecto para sistemas que requieren alta disponibilidad y distribución global.
- Manejo de Datos Semi-Estructurados: Ideal para datos que no siguen una estructura fija y tienden a cambiar.

## Historia de MongoDB

MongoDB fue lanzado en 2009 por la compañía 10gen (hoy MongoDB, Inc.). Inicialmente, buscaba llenar la necesidad de una base de datos orientada a documentos que pudiera manejar datos en un entorno distribuido y escalable, **especialmente útil para las aplicaciones web de alta carga**. La popularidad de MongoDB creció rápidamente en el ámbito del desarrollo de **aplicaciones modernas**.

## Importancia de MongoDB

MongoDB es fundamental en aplicaciones modernas, especialmente aquellas que requieren **adaptarse rápidamente a los cambios de datos y que deben escalar horizontalmente**. Su modelo orientado a documentos y la posibilidad de manejar grandes volúmenes de datos de forma eficiente lo hacen crucial en sistemas **distribuidos**, aplicaciones de **big data** y desarrollos en **tiempo real**.

## 5. ¿Cuándo Usar Bases de Datos Relacionales vs. No Relacionales?

### Usar Bases de Datos Relacionales:

- Cuando la **integridad de los datos y las relaciones son críticas** (ej. sistemas financieros, aplicaciones bancarias).
- Cuando las consultas complejas y las transacciones necesitan cumplir estrictamente las propiedades **ACID**.
- En sistemas de gestión de información estructurada que **no necesitan cambiar su esquema constantemente** (ej. sistemas de CRM o ERP).

### Usar Bases de Datos No Relacionales:

- Para grandes volúmenes de datos no estructurados o semi-estructurados que **no requieren relaciones estrictas** (ej. redes sociales, sistemas de logs).
- En aplicaciones que requieren **escalabilidad horizontal y alta disponibilidad**.
- Para desarrollos ágiles donde los datos cambian con frecuencia o deben adaptarse a **nuevos requisitos rápidamente** (ej. plataformas de comercio electrónico).

MongoDB, en particular, es excelente para aplicaciones en tiempo real, big data, y proyectos de microservicios, donde los datos no necesitan estar altamente estructurados, y se busca rapidez y escalabilidad sobre la consistencia fuerte de los datos.

## Glosario

1. **MongoDB:** Una base de datos NoSQL orientada a documentos, diseñada para ser escalable y flexible, permitiendo almacenar datos en formato **JSON-like**.
2. **Documento:** La unidad básica de almacenamiento en MongoDB, similar a un objeto JSON, compuesto por pares de clave-valor.
3. **Colección:** Un grupo de documentos en MongoDB. A diferencia de las tablas en las bases de datos relacionales, las colecciones no imponen un esquema fijo en los documentos que las componen.
4. **Clave-Valor:** Un par de datos compuesto por una clave y un valor. En MongoDB, los documentos son básicamente colecciones de pares clave-valor. {clave: valor}
5. **Aggregation:** La canalización de agregación es un marco de procesamiento de datos en MongoDB que permite realizar operaciones de agregación en los documentos de una colección. Realizar cálculos/operaciones avanzadas en mongo en lugar de hacerlo en tu aplicación cliente.
6. **Atomicidad:** En MongoDB, las operaciones son atómicas a nivel de un solo documento, lo que significa que una operación de escritura en un documento es una operación indivisible.
7. **Consistencia:** MongoDB ofrece un nivel de **consistencia eventual por defecto**, pero también es posible configurar niveles de consistencia más fuertes, garantizando que todas las lecturas reflejen los datos más recientes.
8. **Durabilidad:** MongoDB garantiza que las operaciones confirmadas se escriban en el almacenamiento persistente y sean inmunes a la pérdida de datos, incluso en caso de fallo del sistema.
9. **Atlas:** MongoDB Atlas es el servicio de base de datos como servicio (DBaaS) ofrecido por MongoDB, que permite implementar, administrar y escalar clústeres de MongoDB en la nube.
10. **Comandos CRUD:** MongoDB utiliza una serie de comandos CRUD (Crear, Leer, Actualizar, Eliminar) para interactuar con los documentos en las colecciones.
11. **Expresiones de Consulta:** MongoDB utiliza expresiones de consulta para filtrar los documentos en una colección según criterios específicos.
12. **Cluster:** Un conjunto de servidores interconectados que trabajan juntos para almacenar y procesar datos en MongoDB. Un cluster puede consistir en nodos primarios, secundarios, árbitros, shards, config servers, y routers, dependiendo de la arquitectura utilizada.