

Taller Mongo

David Santiago Noguera Perez - 827823

Corporación Universitaria Minuto de Dios

Ing. William Alexander Matallana Porras

10-60747: Bases de datos masivas

2025

## Tabla de contenido

Introducción .....	3
Objetivos .....	4
1. ¿Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL? .....	5
2. ¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL? .....	6
3. ¿Cómo se almacena la información en MongoDB y qué formato utiliza? .....	9
4. Explica la diferencia entre JSON y BSON en MongoDB.....	10
5. Estructura de los archivos JSON.....	11
6. ¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad? .....	13
7. Comandos para realizar CRUD en Mongo .....	13
8. Cómo se pueden relacionar datos en Mongo sin usar joins como en SQL .....	14
9. Descargar imagen de Mongo en Docker .....	14
10. Herramientas similares a Workbench para visualizar los datos de Mongo .....	15
Conclusiones.....	16
REFERENCIAS.....	17

## Introducción

MongoDB es una base de datos NoSQL de tipo documental que ha ganado popularidad debido a su flexibilidad, escalabilidad y rendimiento en el manejo de grandes volúmenes de datos no estructurados o semi-estructurados. A diferencia de las bases de datos relacionales tradicionales, como MySQL, MongoDB almacena los datos en documentos similares a JSON (BSON), lo que permite un esquema dinámico y adaptable a las necesidades cambiantes de las aplicaciones modernas. Este taller tiene como objetivo explorar los conceptos fundamentales de MongoDB, sus diferencias con las bases de datos relacionales, y cómo realizar operaciones básicas (CRUD) en este entorno

## Objetivos

1. **Comprender las diferencias entre MongoDB y las bases de datos relacionales:**
  - Identificar las características clave de MongoDB como base de datos NoSQL.
  - Comparar el modelo de datos documental de MongoDB con el modelo relacional de bases de datos como MySQL.
2. **Aprender los conceptos básicos de MongoDB:**
  - Entender qué es una colección, un documento y cómo se almacenan los datos en MongoDB.
  - Diferenciar entre JSON y BSON, y su uso en MongoDB.
3. **Realizar operaciones CRUD en MongoDB:**
  - Practicar comandos para crear, leer, actualizar y eliminar documentos en MongoDB.
  - Explorar cómo se pueden relacionar datos en MongoDB sin usar joins.
4. **Configurar MongoDB en un entorno Docker:**
  - Aprender a descargar y ejecutar una imagen de MongoDB en Docker.
  - Familiarizarse con herramientas gráficas para visualizar y gestionar datos en MongoDB.
5. **Evaluar las ventajas de MongoDB:**
  - Analizar las ventajas de MongoDB en términos de escalabilidad, flexibilidad y rendimiento frente a las bases de datos relacionales.

## Taller Mongo

### 1. ¿Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL?

MongoDB es una base de datos NoSQL (Not Only SQL) orientada a documentos. Fue desarrollada por MongoDB Inc. y lanzada como software de código abierto en 2009. A diferencia de las bases de datos relacionales tradicionales, MongoDB está diseñada para almacenar, consultar y manipular datos estructurados en formato de documentos JSON.

Las principales diferencias con bases de datos relacionales como MySQL son:

- **Estructura de almacenamiento:** MongoDB almacena los datos en documentos BSON (Binary JSON), que son estructuras flexibles similares a objetos JSON. Estos documentos pueden tener campos variables y estructuras complejas anidadas. En contraste, MySQL almacena los datos en tablas compuestas por filas y columnas con estructuras rígidas y predefinidas.
- **Esquema de datos:** MongoDB utiliza un enfoque de "esquema dinámico", lo que significa que los documentos en una misma colección pueden tener diferentes campos y estructuras. No es necesario definir el esquema antes de insertar datos. Por otro lado, MySQL requiere un esquema predefinido, donde cada tabla tiene una estructura fija con tipos de datos específicos para cada columna.
- **Lenguaje de consulta:** MongoDB utiliza un lenguaje de consulta basado en documentos JSON, mientras que MySQL utiliza SQL (Structured Query Language). Las consultas en MongoDB se construyen como documentos JSON que especifican patrones a buscar en los documentos almacenados.
- **Relaciones entre datos:** En MySQL, las relaciones entre tablas se establecen mediante claves foráneas y se consultan utilizando JOIN. MongoDB no implementa JOINS de manera nativa; las relaciones se manejan ya sea incorporando los datos relacionados directamente dentro de documentos (documentos embebidos) o mediante referencias entre documentos (similar a claves foráneas, pero sin integridad referencial automática).
- **Transacciones y ACID:** Tradicionalmente, MySQL ofrece completo soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad). MongoDB, en sus versiones anteriores, tenía limitaciones en transacciones multi-documento, aunque las versiones más recientes (desde la 4.0) han mejorado significativamente el soporte transaccional.

- **Escalabilidad:** MongoDB está diseñado para escalar horizontalmente mediante sharding (fragmentación), distribuyendo automáticamente los datos entre múltiples servidores. MySQL tradicionalmente está optimizado para escalar verticalmente (aumentando los recursos de un solo servidor), aunque también existen soluciones para clustering y replicación.
- **Indexación:** Ambos sistemas soportan índices para mejorar el rendimiento de las consultas, pero los enfoques son diferentes. MongoDB admite diversos tipos de índices, incluyendo índices compuestos, índices de texto completo, índices geoespaciales e índices para arrays y documentos embebidos.
- **Consistencia vs Disponibilidad:** MongoDB prioriza la disponibilidad y la tolerancia a particiones de red (según el teorema CAP), permitiendo configuraciones que sacrifican cierto grado de consistencia inmediata. MySQL tradicionalmente prioriza la consistencia, aunque esto depende de la configuración específica.

## 2. ¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL?

Una colección en MongoDB es el equivalente conceptual a una tabla en una base de datos relacional como MySQL, pero con diferencias fundamentales en su diseño y funcionamiento.

### Colecciones en MongoDB:

Una colección es un grupo de documentos MongoDB. Es el lugar donde se almacenan los documentos relacionados dentro de una base de datos MongoDB. Las colecciones tienen las siguientes características:

- **Esquema dinámico y flexible:** Una colección puede contener documentos con diferentes estructuras. No todos los documentos en una colección necesitan tener los mismos campos, y la estructura de los documentos puede cambiar con el tiempo.
- **Documentos heterogéneos:** Los documentos dentro de una colección pueden tener campos completamente diferentes entre sí. Por ejemplo, en una colección de "usuarios", algunos documentos podrían tener campos como "teléfono" o "dirección" mientras otros no los tienen.
- **Creación implícita:** Las colecciones se crean automáticamente la primera vez que se inserta un documento en ellas, sin necesidad de definirlas previamente.
- **Tipado de datos dinámico:** Los campos en los documentos pueden contener diferentes tipos de datos, incluso si tienen el mismo nombre en diferentes documentos.

- **Documentos anidados y arrays:** Las colecciones pueden almacenar documentos con estructuras complejas, incluyendo arrays y documentos embebidos dentro de otros documentos, facilitando la representación de datos jerárquicos.
- **Sin restricciones de esquema:** No hay restricciones integradas para validar la estructura de los documentos (aunque MongoDB permite definir validadores de esquema opcionales desde la versión 3.2).

### Tablas en SQL (MySQL):

Una tabla en un sistema relacional como MySQL es una estructura de datos organizada en filas y columnas con las siguientes características:

- **Esquema rígido y predefinido:** Todas las filas en una tabla deben seguir exactamente la misma estructura, con columnas predefinidas y tipos de datos específicos para cada columna.
- **Definición explícita:** Las tablas deben ser definidas explícitamente antes de poder almacenar datos, especificando nombres de columnas, tipos de datos, restricciones, etc.
- **Homogeneidad de datos:** Cada fila en una tabla debe tener el mismo conjunto de columnas. Si un dato no está disponible, se utilizan valores NULL, pero la columna debe existir.
- **Tipado estricto:** Cada columna tiene un tipo de datos específico y todas las entradas en esa columna deben adherirse a ese tipo.
- **Relaciones normalizadas:** Las tablas generalmente están diseñadas siguiendo principios de normalización, con relaciones entre tablas establecidas mediante claves foráneas.
- **Integridad referencial:** Las bases de datos relacionales implementan mecanismos para mantener la integridad referencial entre tablas relacionadas.

### Ejemplos para ilustrar las diferencias:

En MySQL (tabla rígida):

```
CREATE TABLE usuarios (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nombre VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL,  
  edad INT CHECK (edad >= 18),  
  fecha_registro DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

En MongoDB (colección flexible):

```
// Primer documento en la colección "usuarios"  
db.usuarios.insertOne({  
  nombre: "Juan Pérez",  
  email: "juan@ejemplo.com",  
  edad: 28,  
  intereses: ["deportes", "música", "viajes"]  
});  
  
// Segundo documento con estructura diferente en la misma colección  
db.usuarios.insertOne({  
  nombre: "María López",  
  email: "maria@ejemplo.com",  
  dirección: {  
    calle: "Av. Principal 123",  
    ciudad: "Madrid",  
    país: "España"  
  },  
  teléfono: "+34612345678"  
});
```

Como se puede ver, en MongoDB ambos documentos pertenecen a la misma colección a pesar de tener estructuras completamente diferentes, algo que sería imposible en una tabla relacional.



### 3. ¿Cómo se almacena la información en MongoDB y qué formato utiliza?

MongoDB almacena la información utilizando un modelo de datos basado en documentos, en un formato llamado BSON (Binary JSON). Este enfoque ofrece una forma flexible y escalable de representar datos, especialmente adecuada para aplicaciones con estructuras de datos complejas o cambiantes.

#### Estructura jerárquica de almacenamiento

La estructura jerárquica de MongoDB para almacenar datos es:

1. **Servidor MongoDB:** Es la instancia del servicio que ejecuta el motor de base de datos MongoDB.
2. **Base de datos:** Un servidor MongoDB puede contener múltiples bases de datos, cada una funcionando como un contenedor de alto nivel para colecciones. Las bases de datos proporcionan un espacio de nombres para organizar los datos y pueden tener diferentes configuraciones de seguridad y almacenamiento.
3. **Colecciones:** Dentro de cada base de datos, los datos se agrupan en colecciones. Como se explicó anteriormente, las colecciones son análogas a las tablas en bases de datos relacionales, pero sin un esquema fijo.
4. **Documentos:** Son las unidades básicas de almacenamiento en MongoDB. Cada documento es una estructura de datos BSON que contiene pares de campo-valor. Los documentos se almacenan dentro de colecciones.
5. **Campos:** Cada documento contiene uno o más campos, que son pares nombre-valor similares a las propiedades de un objeto en programación orientada a objetos.

## 4. Explica la diferencia entre JSON y BSON en MongoDB.

- **JSON:** Formato de texto ligero para intercambio de datos. Es legible por humanos y se basa en pares clave-valor.
- **BSON:** Es una versión binaria de JSON. Es más eficiente para el almacenamiento y procesamiento, ya que admite tipos de datos adicionales (como fechas, binarios, etc.) y es más rápido para ser analizado por máquinas.

### Diferencias clave:

#### 1. Formato:

- JSON es texto plano.
- BSON es binario.

#### 2. Tipos de datos:

- JSON soporta tipos básicos como strings, números, booleanos, arrays y objetos.
- BSON soporta tipos adicionales como fechas, binarios, y ObjectId.

#### 3. Eficiencia:

- BSON es más eficiente para el almacenamiento y procesamiento.
- JSON es más legible para humanos.

## 5. Estructura de los archivos JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y basado en texto que se ha convertido en un estándar ampliamente utilizado para la transmisión y almacenamiento de datos estructurados. La estructura de los archivos JSON sigue un conjunto de reglas sintácticas bien definidas que permiten representar datos de manera jerárquica y organizada.

Un archivo JSON tiene la siguiente estructura básica:

- Claves: Strings que actúan como identificadores.
- Valores: Pueden ser strings, números, objetos, arrays, booleanos o null.
- Objetos: Representados entre {}.
- Arrays: Representados entre [].

### Ejemplos detallados de estructuras JSON

#### 1. Objeto JSON simple

```
{  
  "nombre": "Ana Martínez",  
  "edad": 28,  
  "activo": true,  
  "saldo": 1542.63,  
  "cuenta_verificada": null  
}
```

#### 2. Array JSON simple

```
[  
  "Manzana",  
  "Naranja",  
  "Plátano",  
  "Fresa",  
  "Kiwi"  
]
```

### 3. Objeto JSON con arrays

```
{
  "nombre": "Carlos López",
  "edad": 35,
  "hobbies": ["fotografía", "ciclismo", "ajedrez"],
  "números_favoritos": [7, 13, 42, 99]
}
```

### 4. Objeto JSON con objetos anidados

```
{
  "nombre": "Empresa XYZ",
  "fundación": 2010,
  "ubicación": {
    "calle": "Av. Principal 123",
    "ciudad": "Barcelona",
    "código_postal": "08001",
    "país": "España",
    "coordenadas": {
      "latitud": 41.3851,
      "longitud": 2.1734
    }
  },
  "activa": true
}
```

### 5. Array de objetos JSON

```
[
  {
    "id": 1,
    "nombre": "Smartphone XS2000",
    "precio": 799.99,
    "disponible": true,
    "categoría": "electrónica"
  },
  {
    "id": 2,
    "nombre": "Laptop UltraBook Pro",
    "precio": 1299.50,
    "disponible": false,
    "categoría": "computadoras"
  },
  {
    "id": 3,
    "nombre": "Auriculares Noise Cancel",
    "precio": 249.99,
    "disponible": true,
    "categoría": "accesorios"
  }
]
```

## 6. ¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad?

- **Escalabilidad:** MongoDB está diseñado para escalar horizontalmente (sharding), lo que permite distribuir la carga entre múltiples servidores.
- **Flexibilidad:** Al ser schemaless, permite almacenar documentos con estructuras diferentes en la misma colección, lo que facilita la evolución del modelo de datos.
- **Rendimiento:** Al no usar joins, las consultas pueden ser más rápidas en ciertos escenarios.
- **Almacenamiento de datos no estructurados:** Ideal para datos JSON-like o jerárquicos.

## 7. Comandos para realizar CRUD en Mongo

- **Create (Insertar):**

```
bash

db.coleccion.insert({ clave: "valor" })
db.coleccion.insertOne({ clave: "valor" })
db.coleccion.insertMany([ { clave1: "valor1" }, { clave2: "valor2" } ])
```

- **Read (Consultar):**

```
bash

db.coleccion.find({ clave: "valor" })
db.coleccion.findOne({ clave: "valor" })
```

- **Update (Actualizar):**

```
bash

db.coleccion.update({ clave: "valor" }, { $set: { clave: "nuevo_valor" } })
db.coleccion.updateOne({ clave: "valor" }, { $set: { clave: "nuevo_valor" } })
db.coleccion.updateMany({ clave: "valor" }, { $set: { clave: "nuevo_valor" } })
```

- **Delete (Eliminar):**

```
bash

db.coleccion.remove({ clave: "valor" })
db.coleccion.deleteOne({ clave: "valor" })
db.coleccion.deleteMany({ clave: "valor" })
```

## 8. Cómo se pueden relacionar datos en Mongo sin usar joins como en SQL

En MongoDB, las relaciones se pueden manejar de dos formas:

1. **Documentos anidados:** Incrustar documentos dentro de otros documentos.

```
json
{
  "nombre": "Juan",
  "direccion": {
    "calle": "Calle Falsa",
    "ciudad": "Madrid"
  }
}
```

2. Referencias: Usar un campo para referenciar otro documento en otra colección.

```
json
{
  "nombre": "Juan",
  "direccion_id": ObjectId("1234567890abcdef12345678")
}
```

## 9. Descargar imagen de Mongo en Docker

Para descargar y ejecutar MongoDB en Docker, usa el siguiente comando:

**docker run -d --name mi-mongo -p 27017:27017 mongo**

- -d: Ejecuta el contenedor en segundo plano.
- --name mi-mongo: Asigna un nombre al contenedor.
- -p 27017:27017: Mapea el puerto 27017 del contenedor al puerto 27017 del host.
- mongo: Especifica la imagen de MongoDB.

```
C:\Users\Esteban\Desktop>docker run -d --name mi-mongo -p 27017:27017 mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
5a7813e071bf: Pull complete
d67c4ebf9460: Pull complete
7afa02f8c09e: Pull complete
4e7ca17a42bd: Pull complete
342a4f4728ff: Pull complete
d5bafd14f8e8: Pull complete
0c492c8e8cfd: Pull complete
734719e891c0: Pull complete
Digest: sha256:7bd28e5eea1c5766a084d5818254046f3ebe3b8f20a65e3a274640189e296667
Status: Downloaded newer image for mongo:latest
00e2bc729491da4a773a551e3fc8c89d414ae9879780b9b2287e8d7213128a06
```

## 10. Herramientas similares a Workbench para visualizar los datos de Mongo

- **MongoDB Compass:** Herramienta oficial de MongoDB para visualizar y manipular datos.
- **Robo 3T (Robomongo):** Cliente gráfico ligero para MongoDB.
- **Studio 3T:** Herramienta avanzada con soporte para consultas SQL en MongoDB.
- **NoSQLBooster:** Cliente gráfico con soporte para múltiples bases de datos NoSQL.
- **MongoDB Atlas:** Servicio en la nube de MongoDB que incluye herramientas de visualización.

Estas herramientas permiten explorar colecciones, ejecutar consultas y analizar datos de manera visual.

## Conclusiones

MongoDB es una herramienta poderosa y versátil para el manejo de datos en aplicaciones modernas, especialmente cuando se requiere flexibilidad en el esquema de datos y escalabilidad horizontal. A lo largo de este taller, hemos explorado sus conceptos fundamentales, como colecciones, documentos, y el uso de BSON para el almacenamiento eficiente de datos. También hemos practicado operaciones CRUD y aprendido a configurar MongoDB en un entorno Docker.

Las principales ventajas de MongoDB sobre las bases de datos relacionales incluyen su capacidad para manejar datos no estructurados, su escalabilidad horizontal mediante sharding, y su rendimiento optimizado para consultas en grandes volúmenes de datos. Sin embargo, es importante considerar que MongoDB no es una solución universal; su uso debe evaluarse en función de los requisitos específicos de cada proyecto.



## REFERENCIAS

MongoDB. (s. f.). *MongoDB Atlas: Cloud Document Database*.

[https://www.mongodb.com/es/lp/cloud/atlas/try4?utm\\_source=bing&utm\\_campaign=search\\_bs\\_pl\\_evergreen\\_atlas\\_core\\_prosp-brand\\_gic-null\\_amers-co\\_ps-all\\_desktop\\_es-la\\_lead&utm\\_term=mongodb&utm\\_medium=cpc\\_paid\\_search&utm\\_ad=p&utm\\_ad\\_campaign\\_id=662815053&adgroup=1323814371704586&cq\\_cmp=662815053&msclkid=2b4af15e19eb1e20ac1a9683bf921f61](https://www.mongodb.com/es/lp/cloud/atlas/try4?utm_source=bing&utm_campaign=search_bs_pl_evergreen_atlas_core_prosp-brand_gic-null_amers-co_ps-all_desktop_es-la_lead&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=p&utm_ad_campaign_id=662815053&adgroup=1323814371704586&cq_cmp=662815053&msclkid=2b4af15e19eb1e20ac1a9683bf921f61)

MongoDB. (s. f.-a). *JSON y BSON | MongoDB*.

<https://www.mongodb.com/es/resources/basics/json-and-bson?msckid=1ac65b69b20b609f18614e52b3a46195>

Jesús. (2024, 2 octubre). *¿Qué es el Formato BSON en MongoDB?* Tutoriales Dongee.

<https://www.dongee.com/tutoriales/que-es-el-formato-bson-en-mongodb/>

Com, A. I. (2023, 3 octubre). *MongoDB: Que es, ventajas, como funciona y limitaciones*.

Aprende Informática. [https://aprendeinformaticas.com/mongodb-que-es/#google\\_vignette](https://aprendeinformaticas.com/mongodb-que-es/#google_vignette)

Ventajas y Desventajas. (2024, 5 junio). *Ventajas y desventajas de MongoDB: Análisis*

completo. <https://www.ventajasydesventajas.org/ventajas-y-desventajas-de-mongodb-analisis-completo/>

C, D. (2025, 4 marzo). *¿Relaciones en MongoDB? ¡Sí es posible! Guía Rápida* 🚀 ⚡ 💡.

Medium. <https://medium.com/@diego.coder/relaciones-en-mongodb-edf2107a94ad>

Babic, T. (2022, 21 julio). *Cómo unir tablas en SQL sin usar JOINS*. LearnSQL.es.

<https://learnsql.es/blog/como-unir-tablas-en-sql-sin-usar-joins/>

Robledano, A. (2019, 28 octubre). *Qué es MongoDB*. OpenWebinars.net.

<https://openwebinars.net/blog/que-es-mongodb/>

InnovaciónDigital, R. (2023, 19 julio). *Qué es MongoDB y qué diferencia hay con las bases de*

*datos tradicionales*. *InnovaciónDigital360*. <https://www.innovaciondigital360.com/big-data/que-es-mongodb-y-que-diferencia-hay-con-las-bases-de-datos-tradicionales/>