

# MongoDB: El Latido Rebelde de los Datos

*ROMPE LAS CADENAS DE LOS ESQUEMAS RÍGIDOS*

*La revolución de los datos ya está aquí — y no pide permiso.*

*LIBERA TU INFORMACIÓN DEL TIRANO RELACIONAL*

*Donde las tablas se desvanecen — y los documentos reinan.*

*BIENVENIDO A LA ERA DE LA LIBERTAD DE DATOS*

# EL VIAJE: DESVELANDO EL PODER OCULTO

01 LA ANATOMÍA DE LA BESTIA: CARACTERÍSTICAS DE MONGODB

02 EL CHOQUE DE TITANES: MONGODB VS. SQL

03 TERRENOS DE CONQUISTA: CASOS DE USO REALES

04 EL LENGUAJE DEL PODER: COMANDOS ESENCIALES



# ¿QUÉ ES ESTA BESTIA NOSQL?

## ORIENTADO A DOCUMENTOS

Almacena datos en documentos tipo JSON — liberándote de las restricciones tabulares. Cada documento puede tener su propia estructura única.

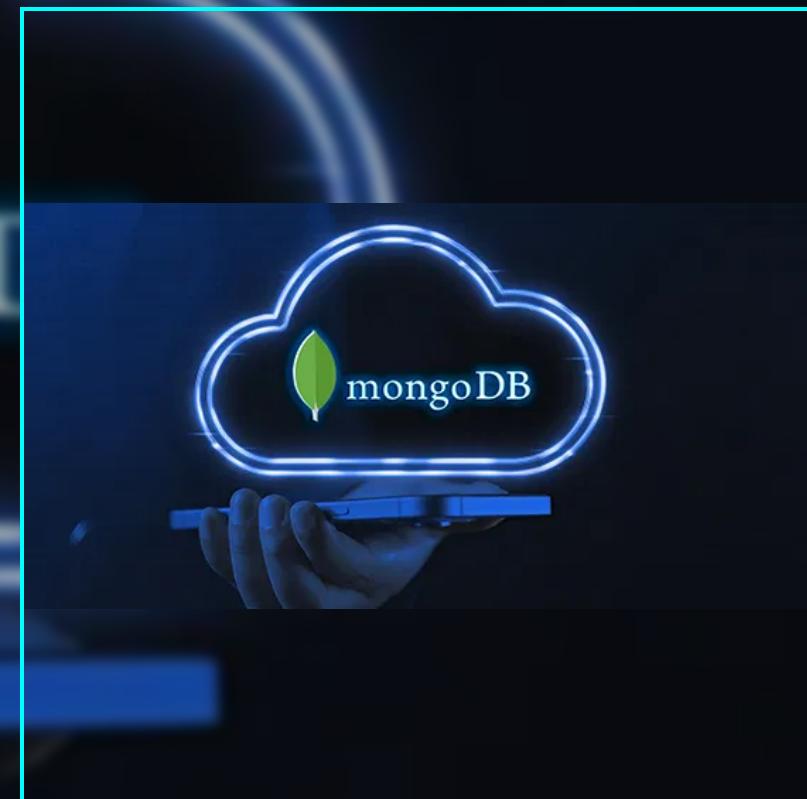
## FLEXIBILIDAD DE ESQUEMA

Sin esquemas predefinidos — los campos pueden variar entre documentos y la estructura puede modificarse sin dolor.

## JSON/BSON

Formato intuitivo para desarrolladores — almacenado como BSON (Binary JSON) para mayor eficiencia y tipos de datos adicionales.

*¿Te atreves a romper el molde?*



# ¡ROMPIENDO ESQUEMAS!

## ESCALABILIDAD HORIZONTAL

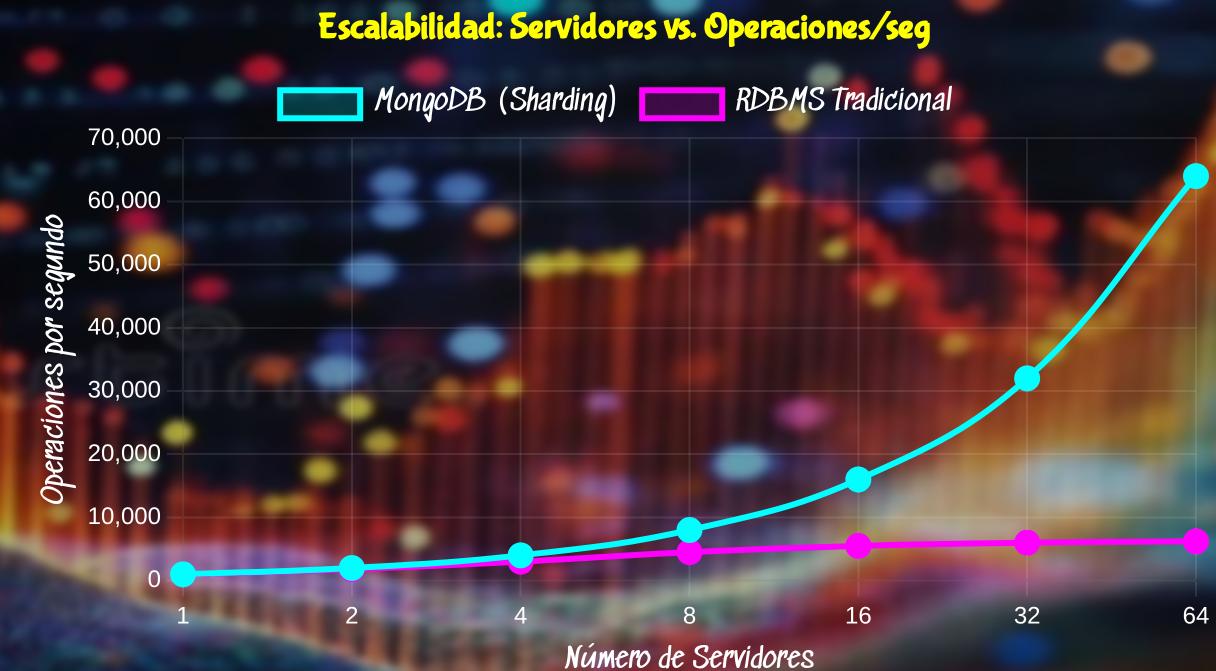
*Sharding automático — distribuye datos entre múltiples servidores sin cambios en la aplicación. Escala a petabytes de datos y miles de operaciones por segundo.*

## ALTA DISPONIBILIDAD

*Replicación nativa con conjuntos de réplicas — recuperación automática y tolerancia a fallos. Garantiza que tus datos siempre estén disponibles.*

## INDEXACIÓN POTENTE

*Soporta índices secundarios, compuestos, geoespaciales, de texto completo y más — optimiza cualquier tipo de consulta para rendimiento máximo.*



**¡LIBERTAD PARA TUS DATOS!**



# ¿GUERRA DE PARADIGMAS?

CONCEPTO	SQL	MONGODB
ESQUEMA	Rígido y predefinido — cambios requieren migraciones dolorosas	Dinámico y flexible — evoluciona con tu aplicación
RELACIONES	JOINS entre tablas — complejidad y sobrecarga de rendimiento	Documentos incrustados — datos relacionados en un solo lugar
TRANSACCIONES	ACID completo — consistencia estricta a costa de disponibilidad	BASE con ACID opcional — equilibrio entre consistencia y disponibilidad

¿ESTÁS LISTO PARA EL CAMBIO?

# EL VIEJO MUNDO VS. LA NUEVA ERA

## ESCALABILIDAD

SQL: Escalabilidad vertical — más potencia en un solo servidor, límites físicos, costos exponenciales.

MongoDB: Escalabilidad horizontal — distribuye la carga entre múltiples servidores, sin límites prácticos, costos lineales.

## MODELADO DE DATOS

SQL: Normalización — fragmenta datos relacionados en múltiples tablas para evitar redundancia.

MongoDB: Desnormalización — mantiene datos relacionados juntos para acceso rápido y directo.

*Donde las tablas se desvanecen, y los documentos reinan.*



# TERRENOS DE CONQUISTA: ¿DÓNDE BRILLA?

## BIG DATA

Gestión de volúmenes masivos de datos — *escalabilidad horizontal sin límites para análisis y procesamiento.*

## INTERNET DE LAS COSAS

Captura de millones de eventos por segundo — *esquemas flexibles para datos de sensores en constante evolución.*

## CMS Y CATALOGOS

Gestión de contenido heterogéneo — *estructura adaptable para diferentes tipos de productos y contenidos.*

## APLICACIONES MÓVILES

Sincronización offline-online — *formato JSON nativo para comunicación directa con frontends.*

## Idoneidad por Caso de Uso

MongoDB SQL Tradicional



¿TU PROYECTO CLAMA POR ESTA FUERZA?



# EL LENGUAJE DEL PODER: TUS PRIMEROS GOLPES

## CONEXIÓN Y NAVEGACIÓN

mongo

Inicia el shell de MongoDB

show dbs

Lista todas las bases de datos

use <database>

Cambia a la base de datos especificada

## OPERACIONES CRUD

db.collection.insertOne({...})

Inserta un documento en la colección

db.collection.find({...})

Busca documentos que coincidan con el filtro

db.collection.updateOne({...}, {...})

Actualiza un documento que coincide con el filtro

db.collection.deleteOne({...})

Elimina un documento que coincide con el filtro



MongoDB Shell

```
> mongo
MongoDB shell version v5.0.6
> show dbs
admin 0.000GB | config 0.000GB | myapp 0.001GB
> use myapp
switched to db myapp
> db.users.insertOne({name: "Ana", age: 28})
"acknowledged": true, "insertedId": ObjectId("60d5...")}
> db.users.find({name: "Ana"})
{ "_id": ObjectId("60d5..."), "name": "Ana", "age": 28 }
> |
```

¡DESATA EL POTENCIAL!

shutterstock.com • 2037430928

# MAESTRÍA EN EL CAMPO DE BATALLA

## COMANDOS AVANZADOS

`db.collection.aggregate([...])`

Pipeline de agregación para análisis y transformación de datos

`db.collection.createIndex({...})`

Crea índices para optimizar consultas

`db.collection.drop()`

Elimina una colección completa

`db.dropDatabase()`

Elimina la base de datos actual

## OPERADORES ÚTILES

`$gt, $lt, $gte, $lte, $ne`

Comparación: mayor, menor, mayor o igual, menor o igual, no igual

`$in, $nin`

Coincidencia con valores en un array o no en un array

`$and, $or, $not`

Operadores lógicos para consultas complejas

## Ejemplo de Agregación

```
1 db.ventas.aggregate([
2 { $match: {
3 fecha: { $gte: new Date('2023-01-01') }
4 }},
5 { $group: {
6 _id: "$categoria",
7 totalVentas: { $sum: "$monto" },
8 promedioVenta: { $avg: "$monto" }
9 }},
10 { $sort: { totalVentas: -1 }}
11 ]) // Ordena por ventas descendente
```

*Cada comando, un paso hacia la dominación.*

shutterstock.com • 1985240786

# EL FUTURO ES NOSQL: ¿TE UNES?

*MongoDB no es solo una base de datos — es una revolución en la forma de pensar sobre los datos. Libera tu creatividad de las restricciones relacionales y abraza la flexibilidad del futuro.*



**VELOCIDAD**  
Rendimiento optimizado  
para operaciones de  
lectura/escritura



**ESCALABILIDAD**  
Creece sin límites con  
sharding automático



**FLEXIBILIDAD**  
Esquemas dinámicos que  
evolucionan con tu  
aplicación

## ¡EL PODER ESTÁ EN TUS MANOS!

[mongodb.com](http://mongodb.com) | [university.mongodb.com](http://university.mongodb.com) | [github.com/mongodb](http://github.com/mongodb)

