



**DESARROLLO WEB**

**BASES DE DATOS NOSQL**

# BASES DE DATOS NOSQL: INTRODUCCIÓN A MONGODB Y SUS CARACTERÍSTICAS

MongoDB representa uno de los sistemas de bases de datos NoSQL más populares y versátiles, diseñado específicamente para aplicaciones modernas que requieren escalabilidad horizontal, flexibilidad de esquema y manejo eficiente de datos semi-estructurados. Como base de datos orientada a documentos, MongoDB almacena datos en formato BSON (Binary JSON), permitiendo estructuras de datos complejas y anidadas que se alinean naturalmente con los objetos utilizados en lenguajes de programación modernos. Esta característica elimina la necesidad de mapping objeto-relacional complejo y facilita el desarrollo ágil de aplicaciones con requisitos de datos evolutivos.

Las características distintivas de MongoDB incluyen escalamiento horizontal automático mediante sharding, replicación integrada para alta disponibilidad, índices flexibles incluyendo geoespaciales y de texto, y un potente lenguaje de consultas que soporta operaciones de agregación complejas. MongoDB implementa el modelo de consistencia eventual por defecto, aunque permite configurar niveles de consistencia más estrictos, según los requisitos de la aplicación. Contreras Castañeda (2016), señala que las aplicaciones web multiplataforma frecuentemente requieren sistemas de almacenamiento que puedan adaptarse rápidamente a cambios en los requisitos de datos, capacidad que MongoDB proporciona mediante su esquema dinámico y herramientas de migración de datos.

En el contexto de la ingeniería de software, MongoDB se utiliza extensivamente en aplicaciones que manejan datos complejos y variados como sistemas de gestión de contenido, aplicaciones IoT, plataformas de redes sociales y sistemas de logging. Empresas como Facebook utilizan MongoDB para almacenar datos de sesiones de usuario, mientras que las startups tecnológicas frecuentemente adoptan MongoDB para acelerar el desarrollo de prototipos y MVPs (Minimum Viable Products) debido a su flexibilidad de esquema y facilidad de desarrollo.

**Ejercicio práctico.** Diseñar e implementar una base de datos MongoDB para un sistema de blog con comentarios anidados.

- Paso 1.** Conectar a MongoDB y crear colección: `const MongoClient = require('mongodb').MongoClient; const client = new MongoClient('mongodb://localhost:27017'); await client.connect(); const db = client.db('blog'); const posts = db.collection('posts');`
- Paso 2.** Insertar post con estructura compleja: `await posts.insertOne({ titulo: 'Introducción a MongoDB', autor: { nombre: 'Juan Pérez', email: 'juan@email.com' }, contenido: 'MongoDB es una base de datos NoSQL...', tags: ['mongodb', 'nosql', 'base-datos'], fecha_creacion: new Date(), comentarios: [{ autor: 'Ana García', texto: 'Excelente artículo', fecha: new Date(), respuestas: [{ autor: 'Carlos López', texto: 'Totalmente de acuerdo', fecha: new Date()} ] }, { autor: 'Carlos López', texto: 'Totalmente de acuerdo', fecha: new Date()} ] }, metadatos: { vistas: 0, likes: 0, compartidos: 0 } });`

**Paso 3.** Consultas complejas: // Buscar posts por tag: `await posts.find({tags: 'mongodb'})`; // Buscar posts con más de 10 comentarios: `await posts.find({'comentarios.10': {'$exists: true'}})`; // Agregación para estadísticas: `await posts.aggregate([{$group: {_id: '$autor.nombre', total_posts: {$sum: 1}, total_vistas: {$sum: '$metadatos.vistas'}}])`;

Resultado. Se obtiene una base de datos funcional que demuestra la flexibilidad de MongoDB para manejar estructuras de datos complejas y consultas avanzadas.