



**Universidad Centroamericana  
“José Simeón Cañas”**

**NoSQL: bases de datos orientadas a documentos sobre el gestor MongoDB**

Ciclo 02/2018

**Autores:**

- |  |          |
|--|----------|
| 1. Eduardo Alberto López Torres        | 00092117 |
| 2. David Alejandro Guardado Chinchilla | 00184217 |
| 3. Mauricio Alejandro Pacheco Guerrero | 00170517 |
| 4. Roberto Ernesto Hernández Contreras | 00162317 |
| 5. Joshua Steven Sharp Reyes           | 00159817 |

**Materia:**

Administración de bases de datos

San Salvador, 08 de noviembre de 2018

# Índice

**Introducción**3

**Marco Teórico**4

**NoSQL**4

**Ventajas de bases NoSQL**4

**Desventajas bases NoSQL**4

**Cuando usar NoSQL**5

**SQL**5

**Ventajas de las bases de datos relacionales**5

**Desventajas bases relacionales**5

**¿Cuándo usar SQL?**5

**Diferencias en SQL y NoSQL.**6

**MongoDB**6

**Principales características de MongoDB:**7

**Ventajas y Desventajas de MongoDB**7

**Conclusiones**8

**Bibliografía**9

**Anexos**10

|

# Introducción

En el presente trabajo, se mostrará la introducción y explicación sobre el lenguaje NoSQL para los Sistema Gestores de Bases de datos, e información y varias funcionalidades que MongoDB trae para su uso en varios programas en los que se puede utilizar. Se detallarán las utilidades, ventajas y desventajas, y su comparación con el lenguaje SQL en MongoDB.

Como primer punto, se definirá el concepto de lenguaje NoSQL, explicando en sí en que consta y sobre que lenguaje se basa. Además, se buscará mostrar todas las utilidades que un lenguaje NoSQL nos puede ofrecer ante el lenguaje SQL, puesto que el lenguaje SQL es más utilizado. El lenguaje NoSQL consigue ofrecer más funcionalidades, las cuales el lenguaje SQL carece.

Luego, se explicará de qué manera el lenguaje NoSQL se almacena. Explicando cada uno de los elementos que esta función compone y como cada uno de estos están compuestos, de manera que sea más sencilla la comprensión de cómo funcionan cada uno de los elementos que este lenguaje nos ofrece a la hora de almacenar los datos.

Después, se explicará el funcionamiento y en que es MongoDB, y poder entender cómo utilizar MongoDB y como este se acopla al lenguaje NoSQL. Además, se explicará cómo MongoDB trabaja de la mano con diversos lenguajes de programación. También se mostrará como MongoDB funciona con lenguajes basados en C, como el caso de JavaScript.

Por último, se mostrarán ejemplos de cómo utilizar MongoDB y como este almacena datos dentro de la base de datos que se creara al momento de exponer la temática. Por medio de estos ejemplos, se buscará que los temas expuestos queden bien comprendidos al terminar este reporte.

# Marco Teórico

## NoSQL

Es un término que describe las bases de datos no relacionales de alto desempeño. Las bases de datos NoSQL utilizan varios modelos de datos, incluidos los de documentos, gráficos, claves-valores y columnas. Las bases de datos NoSQL son famosas por la facilidad de desarrollo, el desempeño escalable, la alta disponibilidad y la resiliencia.

NoSQL, que abarca una amplia gama de tecnologías y arquitecturas, busca resolver los problemas de escalabilidad y rendimiento de big data que las bases de datos relacionales no fueron diseñadas para abordar. NoSQL es especialmente útil cuando una empresa necesita acceder y analizar grandes cantidades de datos no estructurados o datos que se almacenan de forma remota en varios servidores virtuales en la nube.

Contrariamente a las ideas falsas causadas por su nombre, NoSQL no prohíbe el lenguaje estructurado de consultas (SQL). Si bien es cierto que algunos sistemas NoSQL son totalmente no-relacionales, otros simplemente evitan funcionalidades relacionales seleccionadas como esquemas de tablas fijas y operaciones conjuntas.

## Ventajas de bases NoSQL

- La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas.
- Suelen ser bases de datos mucho más abiertos y flexibles. Permiten adaptarse a necesidades de proyectos mucho más fácilmente que los modelos de Entidad Relación.
- Se pueden hacer cambios de los esquemas sin tener que parar bases de datos.
- Escalabilidad horizontal: son capaces de crecer en número de máquinas, en lugar de tener que residir en grandes máquinas.
- Se pueden ejecutar en máquinas con pocos recursos.
- Optimización de consultas en base de datos para grandes cantidades de datos.

## Desventajas bases NoSQL

- No todas las bases de datos NoSQL contemplan la atomicidad de las instrucciones y la integridad de los datos. Soportan lo que se llama consistencia eventual.
- Problemas de compatibilidad entre instrucciones SQL. Las nuevas bases de datos utilizan sus propias características en el lenguaje de consulta y no son 100% compatibles con el SQL de las bases de datos relacionales. El soporte a problemas con las queries de trabajo en una base de datos NoSQL es más complicado.
- Falta de estandarización. Hay muchas bases de datos NoSQL y aún no hay un estándar como si lo hay en las bases de datos relacionales. Se presume un futuro incierto en estas bases de datos.
- Soporte multiplataforma. Aún quedan muchas mejoras en algunos sistemas para que soporten sistemas operativos que no sean Linux.
- Suelen tener herramientas de administración no muy usables o se accede por consola.

## Cuando usar NoSQL

- Redes sociales: casi obligatorio.
- Desarrollo Web: debido a la poca uniformidad de la información que se encuentra en Internet; aun cuando también puede emplearse SQL.
- Desarrollo Móvil: debido a la tendencia – en crecimiento- de Bring Your Own Device.
- BigData: debido a la administración de grandísimas cantidades de información y su evidente heterogeneidad.
- Cloud (XaaS): “Everything as a service”; NoSQL puede adaptarse casi a cualquier necesidad del cliente, y sus particularidades.

## SQL

Las bases de datos relacionales (SQL) son el tipo que más tiempo lleva en el mercado de la tecnología y son por tanto las utilizadas en mayor medida tradicionalmente. Su composición está hecha con bases de datos llenas de tablas con filas que contienen campos estructurados. No se trata precisamente de un tipo de base de datos muy flexible, pero tiene a favor su gran soporte y el enorme desarrollo en herramientas debido a todo su bagaje histórico. El mayor inconveniente de las SQL es probablemente el referido a la escalabilidad, que es precisamente el que permitió el desarrollo de alternativas que desembocaron en las redes NoSQL, ya que grandes empresas que manejan datos a gran escala con necesidad de mucha infraestructura sufrían con este problema.

## Ventajas de las bases de datos relacionales

- Está más adaptado su uso y los perfiles que los conocen son mayoritarios y más baratos.
- Debido al largo tiempo que llevan en el mercado, estas herramientas tienen un mayor soporte y mejores suites de productos y add-ons para gestionar estas bases de datos.
- La atomicidad de las operaciones en la base de datos. Esto es, que en estas bases de datos o se hace la operación entera o no se hace utilizando la famosa técnica del rollback.
- Los datos deben cumplir requisitos de integridad tanto en tipo de dato como en compatibilidad

## Desventajas bases relacionales

- La atomicidad de las operaciones juega un papel crucial en el rendimiento de las bases de datos.
- Escalabilidad, que, aunque probada en muchos entornos productivos suele, por norma, ser inferior a las bases de datos NoSQL.

## ¿Cuándo usar SQL?

- Educación: para estructurar información, y aportar conocimiento lógico al estudiante.
- Desarrollos web: para mantener jerarquía de datos, siempre y cuando la capacidad de concurrencia, almacenamiento y mantenimiento no sean de considerable dificultad y la información sea consistente.
- Negocios: inteligencia y análisis de negocios, son temas que requieren el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- Empresarial: porque tanto el software a la medida y el software empresarial, poseen la característica de mantener información con estructura consistente.

## Diferencias en SQL y NoSQL.

- SQL permite combinar de forma eficiente diferentes tablas para extraer información relacionada, mientras que NoSQL no lo permite o muy limitadamente.
- NoSQL permite distribuir grandes cantidades de información; mientras que SQL facilita distribuir bases de datos relacionales.
- SQL permite gestionar los datos junto con las relaciones existentes entre ellos; en NoSQL no existe este tipo de utilidades.
- NoSQL permite un escalado horizontal sin problemas – por su capacidad de distribución-; mientras que escalar SQL resulta más complicado.

## MongoDB

MongoDB es un sistema de base de datos tipo NoSQL multiplataforma orientado a documentos, de esquema libre. Está escrito en C++, lo que le confiere cierta cercanía al *bare metal*, o recursos de hardware de la máquina, de modo que es bastante rápido a la hora de ejecutar sus tareas. Además, está licenciado como GNU AGPL 3.0, de modo que se trata de un software de licencia libre. Funciona en sistemas operativos Windows, Linux, OS X y Solaris. Su origen se remonta a mediados del año 2000 como parte de un intento de descifrar todos los registros NoSQL llegando espeso y rápido. MongoDB fue de código abierto en el año 2009 y la última versión de MongoDB es 4.0

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección, pueden tener esquemas diferentes.

En MongoDB, cada registro o conjunto de datos se denomina documento. Los documentos se pueden agrupar en colecciones, las cuales se podría decir que son el equivalente a las tablas en una base de datos relacional. Se pueden crear índices para algunos atributos de los documentos, de modo que MongoDB mantendrá una estructura interna eficiente para el acceso a la información por los contenidos de estos atributos. Dichos documentos se almacenan en formato BSON (Binary JSON), la cual es una versión modificada de JSON para realizar búsquedas rápidas de datos. BSON guarda explícitamente las longitudes de los campos, de los arrays y otra información relevante para escaneo de datos. Sin embargo, aunque se guarden en formato BSON, nunca veremos este formato, ya que siempre se trabajarán los documentos en JSON tanto al almacenar como consultar información. JSON es un formato para el intercambio de datos similar a XML, pero su estructura es más simple.

MongoDB viene de serie con una consola desde la que podemos ejecutar los distintos comandos. Esta consola está construida sobre JavaScript, por lo que las consultas se realizan utilizando ese lenguaje. Además de las funciones de MongoDB, podemos utilizar muchas de las funciones propias de JavaScript. En la consola también podemos definir variables, funciones o utilizar bucles.

Si queremos utilizar nuestro lenguaje de programación favorito, existe drivers oficiales para utilizar distintos lenguajes soportados por MongoDB, como por ejemplo los siguientes:

- mongo Shell Edition.
- Node.JS Edition.
- Python Edition.

- C++ Edition.
- Java Edition.
- C# Edition.
- Ruby Edition.

Si bien estos drivers son soportados por MongoDB, no todos están al mismo nivel de desarrollo. Por ende, antes de utilizar estos drivers es mejor revisar su documentación, para ver si son adecuados para el entorno de producción que se desea.

## **Principales características de MongoDB:**

- Cualquier campo en el documento puede ser indexado
- Compatibilidad con la replicación del esclavo maestro.
- La base de datos puede ejecutarse en varios servidores.
- Los datos se duplican para proteger el sistema en caso de falla del hardware.
- Admisión de MapReduce y herramientas de agregación
- Fácil administración en caso de fallas
- Está sin esquema escrito en C ++
- La combinación de MongoDB y JavaScript funciona bien ya que la base de datos usa el lenguaje en lugar de los procedimientos.

## **Ventajas y Desventajas de MongoDB**

### Ventajas

- Plataforma de datos distribuidos: MongoDB se puede ejecutar en todos los centros de datos distribuidos, para garantizar nuevos niveles de disponibilidad y escalabilidad.
- Desarrollo rápido e iterativo: Un modelo de datos flexible con esquema dinámico, con una poderosa interfaz gráfica de usuario y herramientas de línea de comando facilitan a los desarrolladores la creación y evolución de aplicaciones.
- Modelo de datos flexible: Permite el almacenamiento de datos en documentos flexibles similares a JSON, lo que hace que la persistencia de los datos y la combinación sean fáciles.
- TCO reducido (costo total de propiedad): los desarrolladores de aplicaciones pueden hacer su trabajo mucho mejor cuando se usa MongoDB. El equipo de operaciones también puede realizar bien su trabajo gracias al servicio en la nube de Atlas. Los costos se reducen significativamente ya que MongoDB se ejecuta en hardware básico.
- Conjunto de características integrado: se pueden obtener diversas aplicaciones en tiempo real gracias a análisis y visualización de datos, canalizaciones de datos de transmisión por eventos, búsqueda de texto y geoespacial, procesamiento de gráficos, rendimiento en memoria y replicación global de manera confiable y segura.

### Desventajas

- La salida de datos es variable por lo que si necesitas una salida de datos con mas estructura no es conveniente usar mongoDB.
- MongoDB no permite tener una estructura bien definida, por lo que si se necesita de estructura es mejor SQL.
- Aun es una tecnología joven.

## Conclusiones

MongoDB al ser una base de datos orientada a documentos, ofrece una alta disponibilidad, un alto rendimiento y una manera más flexible para la administración de nuestra base de datos, también, al ser un lenguaje NoSQL nos ofrece una manera más sencilla de como estructurar nuestras bases de datos, por lo que es recomendable utilizar MongoDB en cualquier tipo de proyecto, tales como:

- Comercio Electrónico.
- Juegos.
- Aplicaciones móviles.
- Manejo de estadística.
- Administración de contenido.

Proyectos grandes que trabajen BIG DATA

Sin embargo, la implementación de MongoDB no es recomendable para un sistema de transacciones dado que MongoDB no tiene soporte transaccional de manera nativa.



# Bibliografía

(s.f.). Obtenido de <https://docs.mongodb.com/manual/>

(s.f.). Obtenido de <https://docs.mongodb.com/manual/tutorial/manage-users-and-roles/>

(s.f.). Obtenido de <https://docs.mongodb.com/manual/tutorial/enable-authentication/>

(s.f.). Obtenido de <https://docs.mongodb.com/manual/core/authorization/>

(s.f.). Obtenido de <https://docs.mongodb.com/manual/core/security-users/index.html>

<https://expertoenbigdata.com/que-es-mongodb/>

<https://medium.com/@yanyzx/mongodb-qu%C3%A9-es-c%C3%B3mo-funciona-y-cu%C3%A1ndo-podemos-usarlo-8eafe0d441c2>

<https://docs.mongodb.com/manual/release-notes/>

## Anexos

Algunas empresas que utilizan MongoDB.



**BOSCH**



**Expedia®**

**Forbes**

**SEGA**

*Telefónica*

---

Preguntas:

1- ¿En qué año MongoDB fue de código abierto?

- a) 2000
- b) 2001
- c) 2009
- d) 2014

2- ¿Cuál es la última versión funcional de MongoDB?

- a) Versión 3.6
- b) Versión 3.0
- c) Versión 4.0
- d) Versión 4.6

3-El lenguaje NoSQL es característico al no compartir la estructura relacional que un lenguaje SQL. Seleccione una o más:

- A) Los datos se almacenan en filas por medio de key y values.
- B) No permiten operaciones con JOINS.
- C) La expansión de la base de datos de un lenguaje NoSQL es igual de costoso monetariamente que un lenguaje que trabaja con base de datos relacionales.
- D) Su esquema es flexible, es decir, se pueden incorporar nuevos datos sin complicación.
- E) Alberga una gran cantidad de información.

4-Las funciones en MongoDB toman como base la sintaxis de:

- A) JavaScript

- B) Haskell
- C) Oracle Developer
- D) Python

5-Si se tiene la siguiente colección...

_id	gender	age
5beb2431af7a...	male	18.0
5beb2431af7a...	female	20.0
5beb2431af7a...	female	15.0
5beb2431af7a...	male	45.0

¿Cuál debe ser el código para mostrar la cantidad de personas pertenecientes a cada género?

a)

```
db.people.aggregate(
{
  $sort:{
    _id:'$gender',
    totalPeople:{$sum:1}
  }
});
```

b)

```
db.people.aggregate(
{
  $group:{
    _id:'$gender',
    totalPeople:{$sum:2}
  }
});
```

c)

```
db.people.aggregate(
{
  $group:{
    _id:'$gender',
    totalPeople:{$sum:1}
  }
});
```

d)

```
db.people.aggregate(
{
  $sort:{
    _id:'gender',
    totalPeople:{$sum:1}
  }
});
```

6-Si con la anterior colección se desea mostrar la cantidad de personas pertenecientes a cada género, pero con edad mayor a 18 años, Si se tiene la siguiente función map.

```
var map = function () {
  emit(this.gender, { age: this.age, count: 1 });
}
```

¿Cuál debe ser la función reduce?

a)

b)

```

var reduce = function(keys,values) {
  var reduced = {
    totalPeople:0
  }

  for (var i=0; i < values.length; i++) {
    if(values.age >= 18){
      reduced.totalPeople+=1;
    }
  }
  return reduced;
}

```

```

var reduce = function(keys,values) {
  var reduced = {
    totalPeople:0
  }

  for (var i=0; i < values.length; i++) {
    if(values[i].age >= 18){
      reduced.totalPeople+=1;
    }
  }
  return reduced;
}

```

c)

```

var reduce = function(values) {
  var reduced = {
    totalPeople:0
  }

  for (var i=0; i < values.length; i++) {
    if(values.age >= 18){
      reduced.totalPeople+=1;
    }
  }
  return reduced;
}

```

d)

```

var reduce = function(values) {
  var reduced = {
    totalPeople:0
  }

  for (var i=0; i < values.length; i++) {
    if(values[i].age >= 18){
      reduced.totalPeople+=1;
    }
  }
  return reduced;
}

```

7- ¿A qué año se remonta el origen de MongoDB?

a) 2009

b) 2000

c) 1998

d) 2018

8- Escriba en mayúsculas su respuesta. ¿En formato se almacenan los documentos/archivos en MongoDB?

**Respuesta: BSON**

9- ¿En que formato se muestran y trabajan los documentos en MongoDB?

a) JSON.

b) BSON.

c) XML.

d) GRAFOS.

10- ¿Cuál es el lenguaje origen en el cual está escrito MongoDB?

a) JavaScript.

b) Python.

c) Java.

D) C++.