

Base de Datos No SQL

Julio Alfaro

Objetivos de la Semana

- Que el alumno adquiriera el conocimiento de la existencia de otros tipo de Bases de Datos a parte de las Relacionales
- Que el alumno pueda almacenar, modificar y eliminar información en mongoDB
- Que el alumno pueda extraer de MongoDB información para poder de esta forma mostrar los datos que en ella se guardan.
- Entender las diferencias de uso de una base de datos Relacionales vrs una Documental.

Bases de Datos Documentales

Las bases de datos documentales, se conocen así puesto que lo que guardamos son documentos (son elementos JSON o BSON no archivos) y esto es su ventaja lo cual veremos más adelante.

Está enfocada a manejar grandes cantidades de datos y a una gran velocidad, también este tipo de base tiene los siguientes beneficios :

- **De Proposito General** : No esta limitada a un uso específico, incluso podemos usarlo para las mismas funciones que las base de datos Relacional
- **Escalable** : Nos permite crecer y/o mutar nuestra información de manera más sencilla, agregando mas servidores al cluster.

JSON

JSON significa **J**ava**S**cript **O**bject **N**otation, es muy importante tomarla en cuenta pues en nuestro front-end y back-end estaremos utilizando javascript como lenguaje base.

La notación define objetos con los caracteres { }, y via la definicion de una llave y un valor se define cada propiedad, por ejemplo :

```
{ "uno": "valor1", "dos": 123 }
```

En el ejemplo podemos ver un objeto con dos propiedades, una llamada **uno** y otra llamada **dos**, donde el valor de uno es **valor1** y el valor de dos es **123**, y como notamos las hilera deben ir entre comillas los números sin comillas.

JSON

También podemos crear arreglos, que son colecciones de información, y para eso utilizamos los caracteres [], por ejemplo podemos crear un arreglo de números :

[1, 2, 3, 4]

Pero también podemos usar un arreglo de objetos :

[{"tiempo":12}, {"tiempo":15}, {"tiempo":3}]

Y pues estos a su vez son anidables, puedo tener arreglos como valores de una propiedad de objeto o bien objetos dentro de los arreglos.

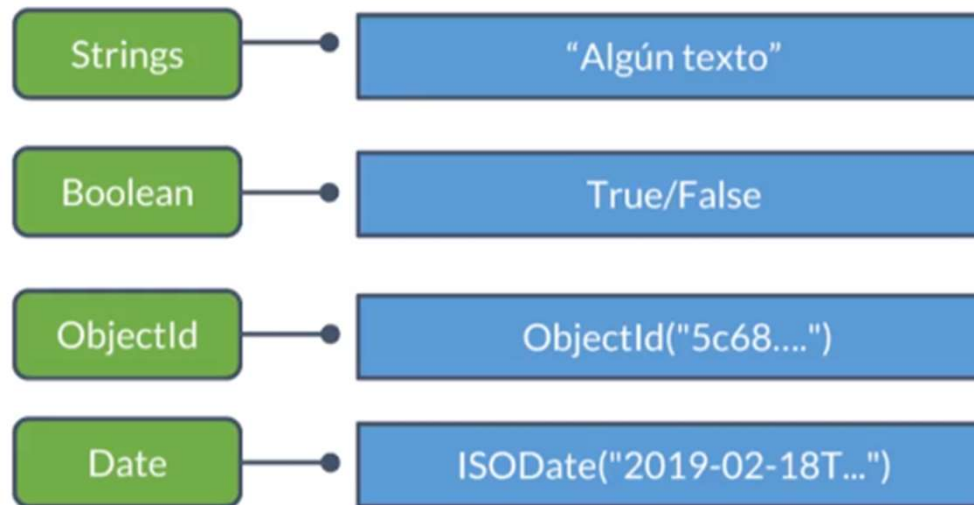
Puedes chequear la página para documentarte mejor : <https://www.json.org/json-en.html>

BSON

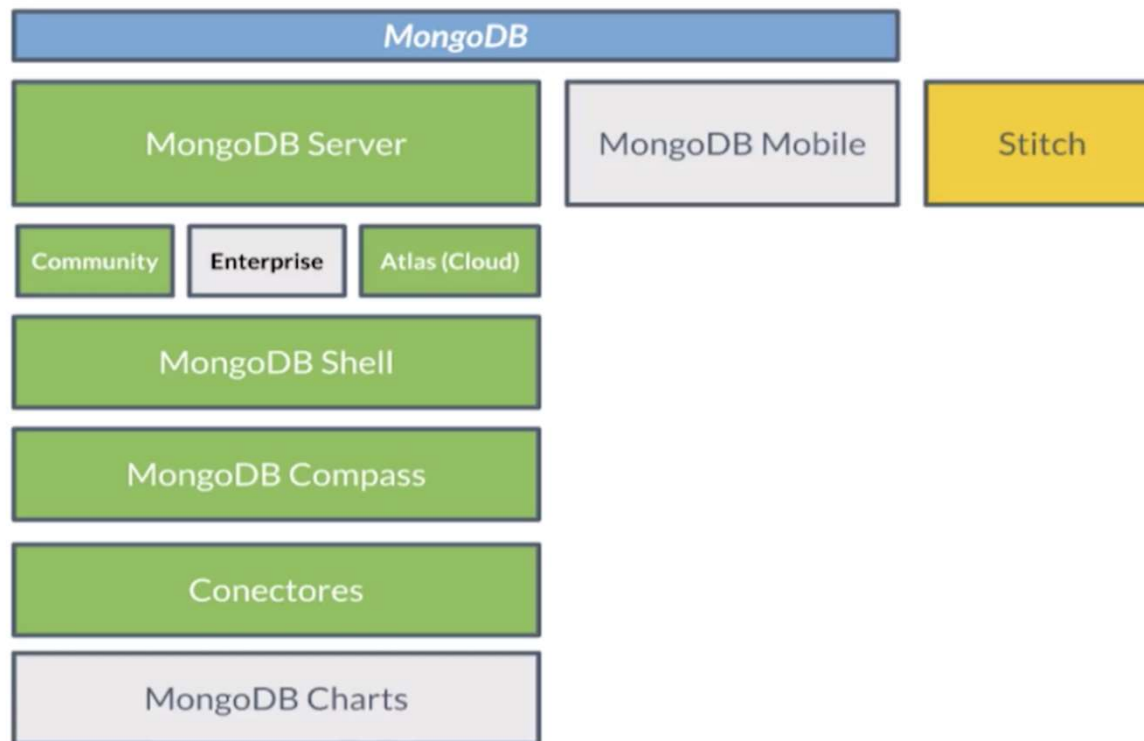
En el caso de los documentos de MongoDB, se utiliza BSON, que es un Binary JSON, el cual su diferencia más grande radica en que interpreta más tipos de datos que un JSON, por ejemplo en JSON, las fechas son una hilera, pero en el BSON, existe el tipo de dato fecha.

Esto es ideal para que tengamos información más certera de los datos que almacenamos en cada documento y podremos hacer cálculos con ellos.

Tipos de Dato



Ecosistema de MongoDB



Uso de MongoDB

Es importante que sepan que pueden instalarlo desde esta dirección :

- **Server :**
<https://www.mongodb.com/try/download/community>
- **Cliente :**
<https://www.mongodb.com/products/compass>

Pero también pueden usar una versión en la nube, que les recomiendo para que la puedan probar, usar esta versión como primera opción :

- <https://www.mongodb.com/cloud/atlas/register>

Estructura de MongoDB

- **La base de datos**

Esta es grupo de unidades lógicas de la información que queramos almacenar, por ejemplo si creó la base de datos universidad, necesitamos almacenar unidades lógicas como estudiantes, maestros, cursos, etc.

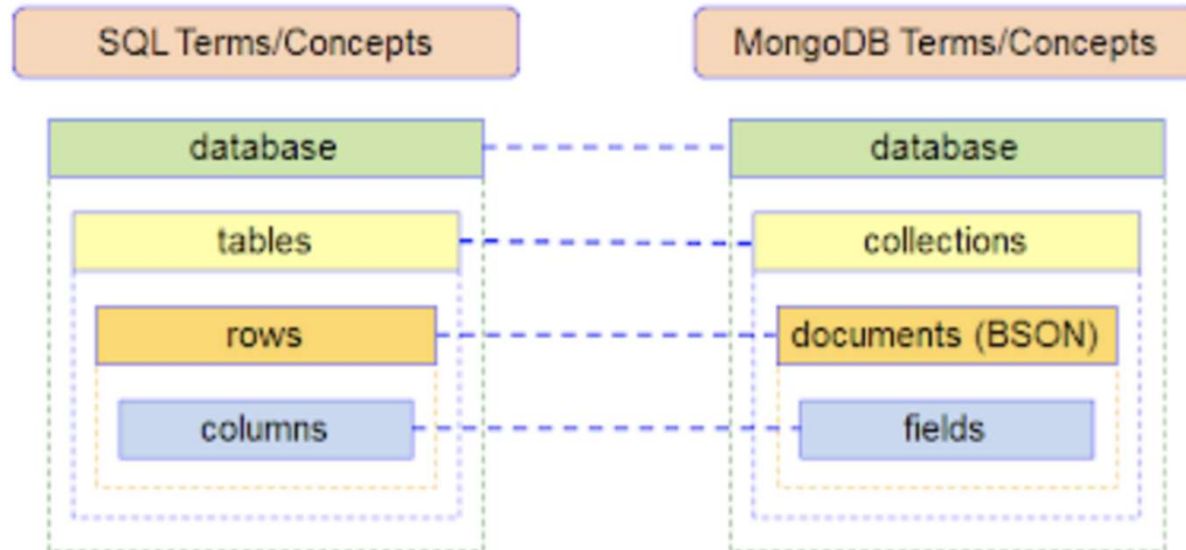
- **Las colecciones**

Este es el nombre que le damos a esas unidades lógicas, por que esta almacena documentos con la información específica, por ejemplo la colección cursos, puede tener Matematica IV, CCIV, Electrónica, etc

- **Los documentos**

Los documentos son la forma de representar cada uno de los datos en las colecciones y usamos JSON para identificarlos.

Equivalencia con estructura de DB Relacional



Comandos importantes

- **db** : Lista la base de datos actual,
- **show dbs** : Lista las bases de dato que tiene nuestro servidor
- **help** : me da la ayuda de los métodos disponibles en la base de datos.
- **use <dbname>** : nos permite crear la base de datos dbname

COMANDOS DE INSERCIÓN DE DATOS

- db.collection.insertOne
- db.collection.insertMany

Comandos importantes

COMANDOS DE ELIMINACIÓN DE DATOS

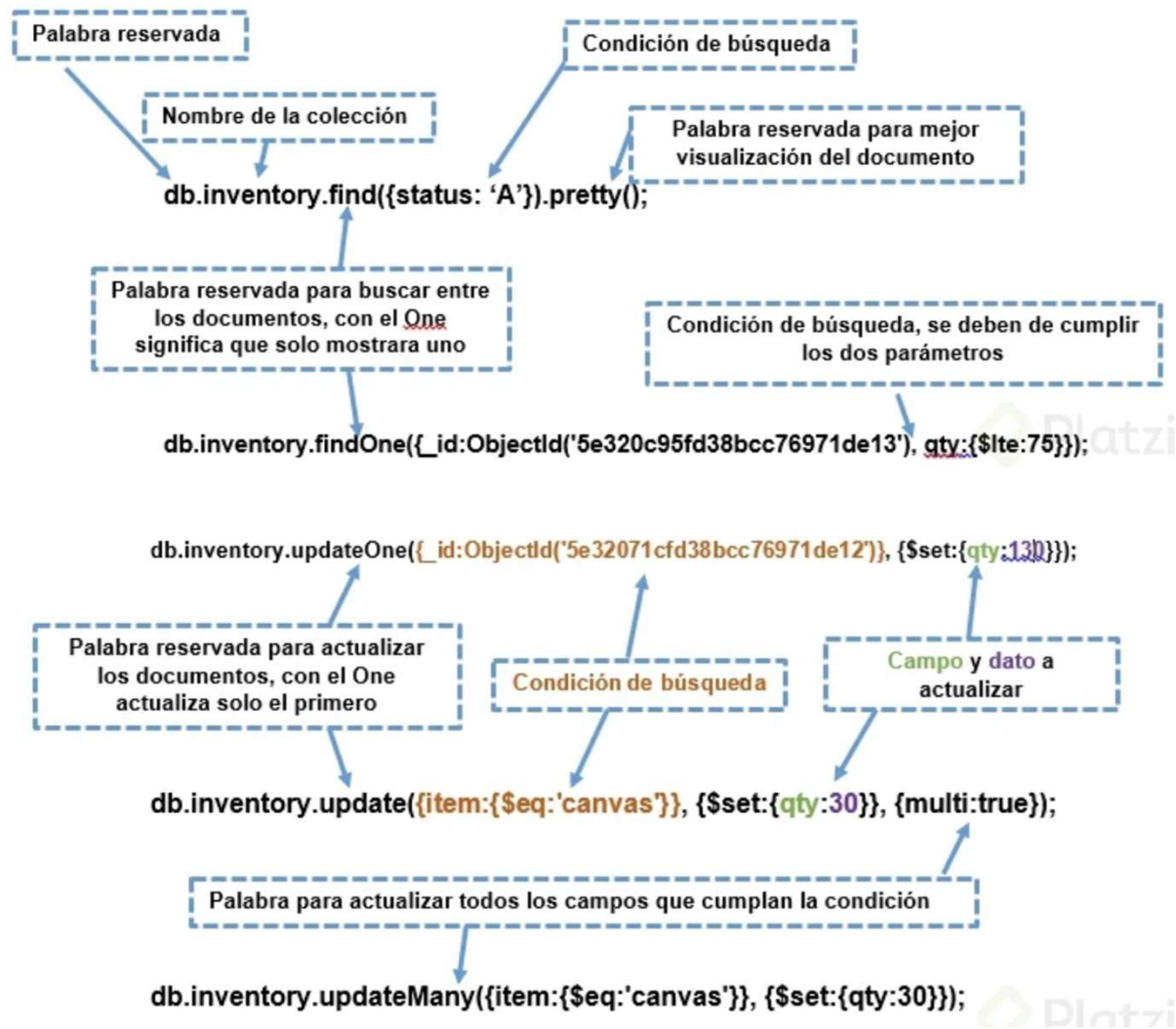
- `db.collection.remove()`

COMANDOS DE MODIFICACIÓN DE DATOS

- `db.collection.update()`
- `db.collection.updateMany()`

COMANDOS DE CONSULTA DE DATOS

- `db.collection.findOne()`
- `db.collection.find()`



Operadores de comparación en MongoDB

\$eq : Permite comparar igualdad	\$lt : Menor que
\$gt : Mayor que	\$lte : Menor e igual que
\$gte : Mayor e igual que	\$ne : No igual
\$in : valida que existan valores en una lista	\$nin : Valida que datos que no existan valores en una lista

Ejemplo de uso `db.collection.find({nota : {$gt : 60}})`

Operadores logicos en MongoDB

\$and : Operador logico Y	\$or : Operador logico O
\$no : Operador logico NO	\$nor : Operador logico

Ejemplo de uso

```
db.collection.find({$and {edad: {$gt : 50}, edad:{$lt : 90 }}});
```


Base de datos NoSQL

DESCRIPCIÓN	MICROSOFT AZURE TABLE STORAGE	AMAZON SIMPLEDDB	GOOGLE CLOUD BIGTABLE
MODELO DE BASE DE DATOS	Wide column store	Wide column store	Key-value store
DESARROLLADOR	Microsoft	Amazon	Google
PUBLICACIÓN INICIAL	2012	2007	2015
ESQUEMA DE DATOS	Libre	Libre	Libre
TIPOS PREDEFINIDOS	Si	no	no
MÉTODOS DE ACCESO	RESTfull API	RESTfull API	gRPC (using protocol buffers) API HBase compatible API (Java)

Tipos de Base de Datos NoSQL



Clave-valor: las bases de datos clave-valor son altamente divisibles y permiten escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar. Los casos de uso como juegos, tecnología publicitaria e IoT se prestan particularmente bien con el modelo de datos clave-valor. [Amazon DynamoDB](#) está diseñado para proporcionar una latencia de milisegundos constante de un solo dígito para cualquier escala de cargas de trabajo. Este rendimiento sistemático es uno de los principales elementos que explican por qué la [característica de historias de Snapchat](#), que incluye la carga de trabajo de escritura de almacenamiento más grande de Snapchat, se trasladó a DynamoDB.



Documentos: en el código de aplicación, los datos se representan a menudo como un objeto o un documento de tipo JSON porque es un modelo de datos eficiente e intuitivo para los desarrolladores. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el uso del mismo formato de modelo de documento que emplean en el código de aplicación. La naturaleza flexible, semiestructurada y jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las aplicaciones. El modelo de documentos funciona bien con catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo. [Amazon DocumentDB \(con compatibilidad para MongoDB\)](#) y MongoDB son bases de datos de documentos conocidas que proporcionan API poderosas e intuitivas para un desarrollo flexible e iterativo.



Gráficos: el propósito de una base de datos de gráficos es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de gráficos incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento. [Amazon Neptune](#) es un servicio de base de datos de gráficos completamente administrado. Neptune admite tanto el modelo de Property Graph como el Resource Description Framework (RDF), que ofrece la opción de dos API de gráficos: TinkerPop y RDF/SPARQL. Las bases de datos de gráficos populares incluyen Neo4j y Giraph.



En memoria: las aplicaciones de juegos y tecnología publicitaria tienen casos de uso como tablas de clasificación, tiendas de sesión y análisis en tiempo real que requieren tiempos de respuesta de microsegundos y pueden tener grandes picos de tráfico en cualquier momento. [Amazon ElastiCache](#) ofrece Memcached y Redis, para servir cargas de trabajo de baja latencia y alto rendimiento, como [McDonald's](#), en las que no se pueden servir con almacenes de datos basados en disco. [Amazon DynamoDB Accelerator \(DAX\)](#) es otro ejemplo de un almacén de datos especialmente diseñado. DAX hace que DynamoDB lea una orden de magnitud más rápida.

Otras bases de Datos NoSQL

Nombre	Descripción
Cassandra	Base de datos creada por Apache del tipo clave-valor. Dispone de un lenguaje propio para realizar consultas CQL (Cassandra Query Language). Cassandra es una aplicación Java por lo que puede correr en cualquier plataforma que cuente con la JVM . Es multiplataforma .
Redis	Base de datos creada por Salvatore Sanfilippo y Pieter Noordhuis y está apoyado por VMWare . Se trata de una base de datos del tipo clave-valor. Se puede imaginar como un array gigante en memoria para almacenar datos, datos que pueden ser cadenas, hashes, conjuntos de datos o listas. Tiene la ventaja de que sus operaciones son atómicas y persistentes. Sin embargo, Redis no permite realizar consultas, solo se puede insertar y obtener datos, además de las operaciones comunes sobre conjuntos (diferencia, unión e inserción). Creado en ANSI C , por lo tanto, es compatible y funciona sin problemas en sistemas Unix , Linux y sus derivados, Solaris , OS/X , sin embargo no existe soporte oficial para plataformas Windows .
MongoDB	Base de datos creada por MongoDB Inc. (anteriormente 10gen) del tipo orientada a documentos, de esquema libre, es decir, que cada entrada puede tener un esquema de datos diferente que nada tenga que ver con el resto de registros almacenados. Es bastante rápido a la hora de ejecutar sus operaciones ya que maneja datos binarios. En poco tiempo, MongoDB se ha convertido en una de las bases de datos NoSQL favoritas por los desarrolladores. Está escrito en lenguaje C++ . Para el almacenamiento de la información, utiliza un sistema propio de documento conocido con el nombre BSON , que es una evolución del formato JSON pero con la peculiaridad de que puede almacenar datos representados de forma binaria. Está disponible para los sistemas operativos Windows , Linux , OS/X y Solaris .
CouchDB	Sistema creado por Apache y escrito en el lenguaje Erlang que funciona en la mayoría de sistemas POSIX (multiplataforma), incluyendo GNU/Linux y OS/X , además de sistemas Windows . Como características más importantes cabe destacar el uso de RESTful HTTP API como interfaz y JavaScript como principal lenguaje de interacción. Para el almacenamiento de los datos se utilizan archivos JSON . Permite la creación de vistas, que son el mecanismo que permite la combinación de documentos para retornar valores de varios documentos, es decir, CouchDB permite la realización de las operaciones JOIN típicas de SQL .