



NoSQL y MongoDB en Español

Autor:

Yohan Graterol B.

5 de diciembre de 2013

A mis padres, hermano y mi novia.

Índice general

I	El desconocido NoSQL	5
1.	Introducción a NoSQL	6
1.1.	Definición	6
1.2.	¿Qué no es NoSQL?	7
1.3.	Tipos de bases de datos no relacionales	8
1.3.1.	Bases de datos orientadas a documentos	8
1.3.2.	Bases de datos orientadas a clave/valor	9
1.3.3.	Bases de datos orientadas a grafos	9
1.4.	Sistema de gestión de bases de datos (SGBD)	10
1.5.	Lista de SGBD NoSQL	10
1.6.	¿Por qué NoSQL?	12

II	Primeros pasos con MongoDB	13
2.	Conociendo MongoDB	14
2.1.	MongoDB	14
2.2.	Términos básicos entorno a MongoDB	15
2.2.1.	JSON - JavaScript Object Notation	15
2.2.2.	Documento	15
2.2.3.	Colección	15
2.3.	Instalando MongoDB	16
2.3.1.	Instalación en Linux desde la fuente	16
2.3.2.	Instalación en Linux desde los repositorios	16

Prefacio

Parte I

El desconocido NoSQL

Capítulo 1

Introducción a NoSQL

1.1. Definición

NoSQL o "No solamente SQL" (**Not Only SQL**) es un término acuñado por Carlo Strozzi en 1998 y nuevamente retomado por Eric Evans en 2009 y se refiere a un conjunto de bases de datos que se diferencian en gran parte de las bases de datos convencionales, en características tanto de uso como de implementación; estos tipos de bases de datos no usan SQL o al menos no como lenguaje predeterminado para realizar las consultas. Las bases de datos NoSQL (desde ahora "no relacionales"), no soportan totalmente **ACID**¹, esto lo explica el teorema del profesor Eric Brewer, Teorema CAP (2000):

Es imposible para un sistema distribuido garantizar simultáneamente las siguientes tres características:

- Consistency (Consistencia): todos los nodos ven la misma data al mismo tiempo.
- Availability (Disponibilidad): una garantía de que todos los requerimientos recibirán una respuesta de que el requerimiento fue exitoso o fallido.
- Partition Tolerance (Tolerancia a la Partición): el sistema continúa operando a pesar de la pérdida arbitraria de mensajes, o

¹'ACID: Atomicidad, Coherencia, Aislamiento y Durabilidad'

CAPÍTULO 1. INTRODUCCIÓN A NOSQL

la falla de parte del sistema.

En primera instancia es una desventaja, pero gracias a esto permite que los motores de bases de datos no relacionales escalen fácilmente de manera horizontal. Para subsanar el problema de ACID, nuevamente el profesor Brewer ideó BASE (Basically Available, Soft-state, Eventually consistent) que lo conforman los siguientes puntos:

- **Disponibilidad básica:** para cada solicitud se garantiza una respuesta, satisfactoria o falla de ejecución.
- **Estado "Soft":** El estado del sistema puede cambiar con el tiempo, a veces sin cualquier entrada.
- **Consistencia Eventual:** La base de datos puede ser en un momento inconsistente pero será consistente con el tiempo.

El lenguaje SQL no es un lenguaje predominante entre los distintos tipos de bases de datos no relacionales, por lo general cada motor tiene su propio lenguaje de consultas. Cabe destacar que la información no se almacena con un esquema fijo (**pero si usando almacenamiento estructurado**), aun que si existe un esquema que el DBA² o el desarrollador propone con anterioridad de manera virtual, es decir, no se crea en el motor antes de utilizar la base de datos sino al almacenar el primer valor.

1.2. ¿Qué no es NoSQL?

NoSQL no es una base de datos y tampoco un tipo de base de datos, sino una definición que engloban un conjunto de tipos de bases de datos que difiere con las bases de datos convencionales.

²'Database Administrator - Administrador de base de datos'

1.3. Tipos de bases de datos no relacionales

En el mundo de las bases de datos no relacionales nos encontramos con distintos modelos o tipos, que se desempeñan mejor en algunos ambientes específicos; esas distintas facetas no se ven en las base de datos relacionales. En este libro se expondrán los tipos más comunes.

1.3.1. Bases de datos orientadas a documentos

Las bases de datos orientadas a documentos o también denominadas como **Bases de datos documental**, trabajan bajo el marco de la definición de un "*Documento*", donde cada motor que usa esta definición difiere en los detalles, pero la mayoría concuerda en como se almacena la información con algún formato estándar. Los formatos más utilizados por los motores más populares son: **JSON** y **BSON**. Se podría considerar este tipo como el más utilizada en la actualidad.

Cada documento, es muy similar a un registro en una base de datos relacional, donde se puede observar un esquema parecido mas no rígido. Dos documentos no tienen porque tener un esquema igual, aunque sean de una misma colección de datos.

```
{
  _id: 1,
  nombre: "MongoDB",
  url: "http://www.mongodb.org",
  tipo: "Documental"
}
```

Figura 1.1: Ejemplo de documento.

Este ejemplo demuestra la sencillez de un documento, se observa un modelo al estilo *clave : valor*. Una analogía con las bases de datos relacionales sería: Clave = Campo y Valor = Dato del campo, hasta allí queda la analogía.

1.3.2. Bases de datos orientadas a clave/valor

Este tipo de bases de datos es muy similar a las bases de datos documental en el concepto de guardar la información con el modelo clave:valor, la diferencia radica en que un documento se almacena en una clave; esta definición puede parecer algo abstracta. Esto se explica mejor con un ejemplo.

El siguiente ejemplo utiliza el documento de la sección anterior:

```
mongodb => {  
  _id: 1,  
  nombre: "MongoDB",  
  url: "http://www.mongodb.org",  
  tipo: "Documental"  
}
```

Figura 1.2: Ejemplo de un documento en una clave.

La clave en este caso es 'mongodb' y su contenido es el mismo documento de la sección anterior. Esto hace que varíe la forma de recuperar la información con respecto a las bases de datos basadas en documentos.

Algun muy interesante de este tipo es que permite ser utilizado junto bases de datos orientadas a documentos, lo que origina motores híbridos.

1.3.3. Bases de datos orientadas a grafos

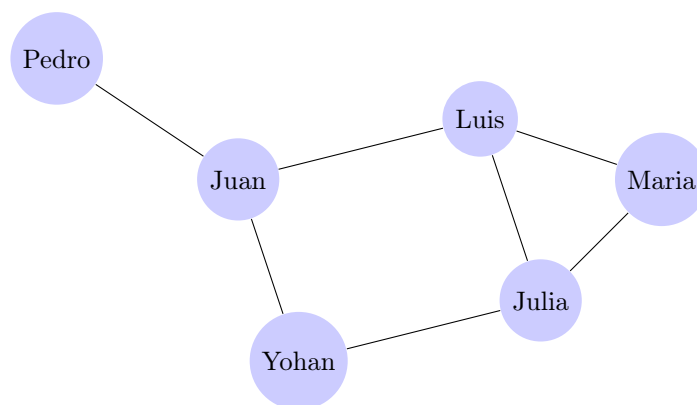


Figura 1.3: Ejemplo de un grafo con relaciones de conocidos.

CAPÍTULO 1. INTRODUCCIÓN A NOSQL

Este tipo difiere completamente a los tipos antes mencionados, y trata la información de una manera peculiar usando **grafos**³ y **teoría de grafos**. Cada nodo solo debe contener una sola columna, por lo tanto se debe normalizar completamente las bases de datos. Y como la definición de grafos indica, las relaciones solo pueden ser binarias, es decir, un nodo puede solo usar una relación para entrar en contacto con otro nodo y no más de uno.

Las ventajas de este tipo de bases de datos van enfocadas a la integridad de los datos, cualquier cambio en un nodo o relación solo afecta localmente.

1.4. Sistema de gestión de bases de datos (SGBD)

Jorge Sánchez Asenjo (2005) define SGBD⁴ como:

Un sistema gestor de bases de datos o SGBD es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

Un tipo de base de datos no sirve de nada sino tiene un sistema que lo gestione, a menos que desees crear un SGBD. En NoSQL hay una basta gama de SGBD, y la mayoría están bajo licencia de código libre, permitiendo así usar, estudiar, modificar y redistribuir sin problema alguno con respecto a algunos motores de bases de datos relacionales con licencias privativas.

1.5. Lista de SGBD NoSQL

Bases de datos documental

- MongoDB
 - Lanzamiento: 2008
 - Licencia: GNU AGPL v3.0

³Grafo: es un conjunto de objetos llamados nodos unidos por enlaces denotados aristas, que permiten representar relaciones binarias entre elementos de un conjunto.

⁴En inglés DBMS: Data Base Management System

CAPÍTULO 1. INTRODUCCIÓN A NOSQL

- CouchDB
 - Lanzamiento: 2005
 - Licencia: Apache License 2.0
- Raven DB
 - Lanzamiento: 2010
 - Licencia: GNU AGPL v3.0

Bases de datos clave/valor

- Apache Cassandra
 - Lanzamiento: 2008
 - Licencia: Apache License 2.0
- Riak
 - Lanzamiento: 2009
 - Licencia: Apache License 2.0
- Redis
 - Lanzamiento: 2009
 - Licencia: BSD

Bases de datos en grafos

- Neo4j
 - Lanzamiento: 2009
 - Licencia: GNU AGPL v3.0
- Dex
 - Lanzamiento: 2008
 - Licencia: Comercial
- Sones GraphDB
 - Lanzamiento: 2012
 - Licencia: GNU AGPL v3.0 y comercial

1.6. ¿Por qué NoSQL?

En esta época donde se generan cantidades enormes de datos menos estructurados, las bases de datos relacionales empiezan a mostrar deficiencias, en almacenamiento u operaciones; siendo esta una de las principales razones de impulsar el uso de bases de datos no relacionales. Muchas personas se quejan del movimiento NoSQL, más que todo por una resistencia al cambio que por los "contras" de este tipo de bases de datos; en la actualidad gestionar una cantidad de datos gigantesca no es tan sencillo si piensas en estructuras, esto da pie al "BigData". Otra de las razones relevantes es la arquitectura, que permite escalar horizontalmente de manera sencilla sin tantos problemas de rendimiento. **En capítulos posteriores veremos que es escalamiento horizontal.**

La velocidad de desarrollo y la velocidad de la base de datos son puntos a favor para las bases de datos no relacionales, reduciendo el tiempo de desarrollo evitando complejas sentencias SQL y además aumentando la velocidad de respuestas para los clientes.

Parte II

Primeros pasos con MongoDB

Capítulo 2

Conociendo MongoDB

2.1. MongoDB

MongoDB¹ es un sistema de bases de datos no relacionales, multiplataforma e inspirada en el tipo de bases de datos documental y clave/valor, su nombre proviene del término en inglés "**humongous**". Está liberada bajo licencia de software libre, específicamente GNU AGPL 3.0². MongoDB usa el formato BSON (JSON Compilado) para guardar la información, dando la libertad de manejar un esquema libre. Este motor de bases de datos es uno de los más conocidos y usados, pudiendolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales.

El desarrollo de MongoDB comenzó en el año 2007 por la empresa 10gen³, publicando una version final en el 2009. Para la fecha que es escrito este libro, MongoDB se encuentra en la versión 2.4.8.

¹MongoDB - <http://www.mongodb.org/>

²AGPL - <http://www.gnu.org/licenses/agpl-3.0.html>

³10gen - <http://www.mongodb.com/>

2.2. Términos básicos entorno a MongoDB

2.2.1. JSON - JavaScript Object Notation

JSON es formato compacto de representacion de objetos. Las especificaciones las publicó Douglas Crockford en el documento RFC 4627⁴. JSON es un formato independiente del lenguaje, aunque su uso extendido hasta hace poco era en el lenguaje Javascript. Actualmente se usa JSON en gran cantidades de sistemas para intercambiar información por su simplicidad en comparación con XML.

Este formato soporta gran cantidad de tipos de datos, lo que lo hace atractivo para un uso generalizado, y cada vez más lenguajes de programación dan soporte a este formato. El ejemplo del capítulo anterior, donde se mostraba un "documento", no es más que JSON.

2.2.2. Documento

Un documento es un conjunto de datos estructurados (mas no con un esquema estricto), que contiene pares clave/valor. Un documento puede ser comparado con una fila o registro en una base de datos relacional.

2.2.3. Colección

Es un conjunto de documentos, similar a una tabla en las bases de datos relacionales.

⁴RFC 4627 - <http://www.ietf.org/rfc/rfc4627.txt>

2.3. Instalando MongoDB

2.3.1. Instalación en Linux desde la fuente

Existen distintas formas de instalar MongoDB en Linux, una de ellas, y la menos recomendable es compilar el código fuente que pueden descargar desde: <http://www.mongodb.org/downloads>. También se puede descargar los binarios, descomprimirlos y usarlos.

2.3.2. Instalación en Linux desde los repositorios

Los sistemas Linux a diferencia de otros SO, maneja sus software en repositorios, que no es más que un sitio centralizado donde se almacenan todos los software disponible para una distribución de Linux.

Instalación en Fedora Linux, Red Hat Linux Enterprise y Derivados

Para instalar MongoDB en alguna de estas distribuciones, se debe hacer uso del gestor de paquetes yum y ejecutar el siguiente comando:

```
yum install mongodb mongodb-server
```

- ***mongodb***: Contiene todos los paquetes “cliente”, como es el caso del cliente **mongo**, la herramienta para respaldos en binarios de bases de datos **mongodump**, **mongorestore** para recuperar respaldos en binario, **mongoexport** y **mongoimport** que realizan una acción similar a mongodump y mongorestore, pero usan formato JSON o CSV.
- ***mongodb-server***: Contiene todos los paquetes para hacer funcionar el servidor, como el demonio **mongod**.

mongodb y mongodb-server se pueden instalar independientemente. ¿Cuándo hacer eso? Un caso muy representativo es cuando se desea colocar en pro-

CAPÍTULO 2. CONOCIENDO MONGODB

ducción la base de datos. Se recomienda solo instalar los paquetes de servicio mongodb-server y el cliente en otra máquina.

Para iniciar/reiniciar o apagar el demonio de MongoDB en **Fedora** se utiliza el siguiente comando:

```
systemctl start|restart|stop mongod.service
```

En el caso de **Red Hat Enterprise Linux 6** o derivados:

```
service mongod start|restart|stop
```