Bases de datos NoSQL en Big Data

Monografía para optar por el título de ingeniero de sistemas

Juan Diego Tovar Ortiz

Código: 066112038

Universidad Libre de Colombia

Facultad de ingeniería

Bogotá D.C.

8 de Septiembre de 2017

RESUMEN

En este documento se pretende exponer al lector los antecedentes, conceptos, las características en base de arquitectura como diseño pertenecientes a las bases de datos NoSQL, como también las múltiples clasificaciones que pueden tener una base de datos de esta índole cuales nos permiten facilitar el uso de análisis en datos en big data también se hará una introducción breve de una de esas bases de datos, MongoDB, con la intención de que el lector pueda tener un claro entendimiento de las fuerzas y debilidades de varias de sistemas de Bases de datos NoSQL para implementar en soluciones de organizaciones .

Palabras Claves: Bases de datos no Relacionales, Bases de datos Relacionales, Teorema de cap, bases de datos clave-valor, bases de datos basadas en grafos, escalabilidad, NoSQL, ACID, BASE, teorema de CAP.

INTRODUCCION

NoSQL, siglas que significan “Not Only SQL” se refiere a un grupo de sistemas de bases de datos no relacionales; cuya característica principal es que no se encuentran construidas en tablas y generalmente no se usa lenguajes comunes del SQL para manipular los datos [1]. Los sistemas de bases de datos en NoSQL también por su naturaleza tienen la utilidad en la facilidad de trabajar con grandes cantidades de datos, uno de los requerimientos necesarios para la analítica de datos propuesto por el Big data[2].

Los sistemas NoSQL se encuentran diseñados para el almacenamiento de datos en larga escala y procesamiento de datos en paralelo a través de varios servidores, son usados por varias compañías reconocidas en internet como Google, Amazon y Facebook, cuyos retos yacían en almacenamiento de grandes cantidades de datos los cuales las RDBMS (Relational Data Base Management Systems) no podían soportar debido a sus diseños. Los sistemas NoSQL pueden

soportar múltiples actividades de consulta y análisis de tipo predictivo como exploratorio.

Los sistemas NoSQL se encuentran diseñadas de acuerdo por los parámetros establecidos por el ACID (Atomicidad Consistencia Aislamiento Durabilidad), BASE (Basic Availavility Soft state Eventual consistency), OLTP (Online Transaction Processing) en tiempo real y la solución OLAP (On-Line Analytical Processing) garantizando la necesidad de las organizaciones en sistemas que puedan almacenar y trabajar datos en cantidades descomunales [3].

Es diversa la aplicabilidad de las bases de datos, entonces en este documento intentaremos resolver las siguientes preguntas ¿Qué modelo de bases de datos se adaptan a determinado tipo de industria? ¿Son las bases de datos NoSQL una evolución de las bases de datos relacionales? ¿Qué características debemos tener en cuenta en una base de datos NoSQL?

DESARROLLO

ANTECEDENTES

De los muchos diferentes modelos de datos el modelo relacional han sido populares desde los años 80s con la implementación del SQL (Structured Query Language), el SQL es un leguaje basado en consultas de acceso a bases de datos relacionales que permite efectuar consultas con el fin de recuperar información de interés de una base de datos como hacer cambios o borrados de forma sencilla[4].

Con implementaciones de Oracle, MySQL y los servidores SQL Microsoft, todos pertenecientes a los RDBMS en la época de los 90s y el 2000 fueron herramientas estándar para una gran cantidad e industrias[5].

Aunque en la actualidad el uso de los bases de datos relacionales se les dificulta en las necesidades que se deben tener las bases de datos hoy en día, en

aspecto como por ejemplo la escalabilidad en varios servidores con grandes cantidades de datos. Esta tendencia se debe a dos razones:

1.El crecimiento exponencial en el volumen de datos generados por los usuarios, sistemas y sensores, acelerado aún más por la concentración de este volumen por sistemas distribuidos como Amazon, Google y varios servicios en la nube.

2.La creciente independencia y complejidad de los datos acelerados por la internet, la Web 2.0, las redes sociales y la necesidad de un acceso estandarizado a fuentes de datos de varios sistemas distintos.

El termino NoSQL que fue acuñado en 1998, originalmente se refería una base de datos relacional de código abierto que no usaba un lenguaje de consultas SQL, al principio fue un concepto el cual no lo tomaron en cuenta pero conforme al paso del tiempo en el año 2009 un empleado de Rackspace Rick Evans reintrodujo el termino en evento el cual discutía las bases de datos de código abierto.

En la actualidad con las organizaciones que recolectan grandes cantidades de datos sin estructurar se encuentran migrando hacia el uso de bases de datos no relacionales, también conocidos como bases de datos NoSQL, las bases de datos NoSQl se enfocan e en los procesos analíticos de datasets a larga escalan ofreciendo escalabilidad en campos como la analítica en Big Data, Inteligencia de negocios y Social Networking.

Características de las bases de datos NoSQL

Para garantizar la integridad de los datos, la mayoría de los sistemas de bases de datos están basados en transacciones. Esto asegura la consistencia de los datos en todas las situaciones de manejo de datos. Estas características

transaccionales son también conocidas como ACID. Aunque todas las métricas propuestas por el ACID, no todas pueden ser cumplidas en los sistemas de bases de datos, por eso se formula teorema de CAP como también BASE.

TEOREMA DE CAP

Según el teorema de CAP o teorema de Brewer formulado por el profesor Eric Brewer en el año 1999 [6], las bases de datos solo pueden garantizar dos de tres características:

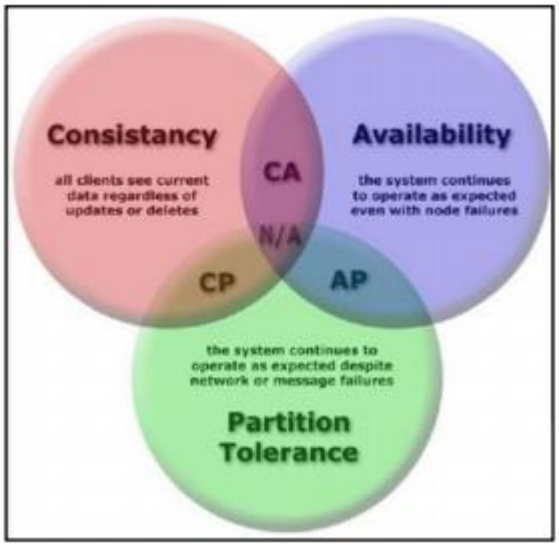

Figura 1. Teorema de CAP

Consistencia: todos los usuarios pueden ver la misma versión de los datos incluyendo actualizaciones o cambios en los datasets.

Availability (disponibilidad): todos los clientes pueden siempre encontrar al menos una copia de los datos solicitados incluso si uno de las maquinas en un clúster se encuentra caído.

Partition Tolerance (Tolerancia a las particiones): los sistemas en su integridad mantienen sus características incluso cuando se encuentra desplegados en diferentes servidores, transparente al cliente.

El teorema de cap postula que solo dos de los tres diferentes aspectos de escalamiento pueden lograrse el mismo tiempo, por tanto un sistema distribuido podrá asegurarnos:

CP: el sistema ejecuta las operaciones de forma consistente aunque se pierda la comunicación entre nodos, pero no asegura que el sistema responda.

AP: el sistema siempre responderá a las peticiones, aunque se pierda la comunicación entre nodos, pero los datos procesados podrían faltarle la consistencia.

CA: el sistema siempre responderá las particiones y los datos procesados serán consistentes, pero en este caso no se permite una pérdida de comunicación entre nodos.

BASE y ACID

Los dos modelos de consistencia en las bases de datos más conocidos son ACID y BASE, ambas explican las métricas que se deben tener al realizar una transacción (consulta, actualizaciones, borrar) y están enfocadas en las bases de datos relacionales (ACID) como las bases de datos no relacionales (Base)[7]

Según ACID transacciones de datos deben ser:

Atómico: todo el contenido de una transacción deberá llevarse a cabo y tener éxito, en caso contrario no se habrá cambios en la base de datos.

Consistente: nada en la realización de la transacción va a dejar la base de datos en un estado inconsistente.

Isolated (aislado): las transacciones no pueden interferir una de la otra, operaran de manera independiente una de la otra.

Durable: las transacciones completadas deben persistir, incluso cuando los servidores se reinicien, también implica que los cambios hechos en la base de datos con la transacción no se podrán deshacer.

El modelo BASE posee un enfoque cercano al ACID solo que en este permite mayor disponibilidad, perdiendo la consistencia y el aislamiento, a favor de la disponibilidad y el rendimiento[7].

Basically Available: hace que el sistema aun funcione a pesar de que alguna parte falle debido a que el almacenamiento sigue los principios de distribución y replicación. Los datos no pueden ser correctos, pero aún puede dar y aceptar respuestas

Soft state (Estado suave): los datos se encuentran en un constante estado de cambio, eso significa que los nodos no tienen que estar consistentes entre si todo el tiempo

Eventual consistency: los datos eventualmente serán consistentes entre todos los nodo en todas las bases de datos, pero no a cada transacción en todo momentos. Llegará con el tiempo a un estado consistente.

Las bases de datos NoSQL se reconocen en cuatro tipos:

Bases de datos basadas en columnas

Como las bases de datos basadas en documentos, las bases de datos basadas en columnas emplea una estructura de datos orientada en columnas en vez de filas, la cual acomoda varios atributos por llaves, los datos son separados y almacenados en familias de columnas, cada columnas es un índice para búsqueda y todos los datos en la columna tienen un mismo tipo para acceso concurrente, todo con el objetivo de que en una consulta sea más fácil y que puedan ser realizados en un mismo proceso[8].

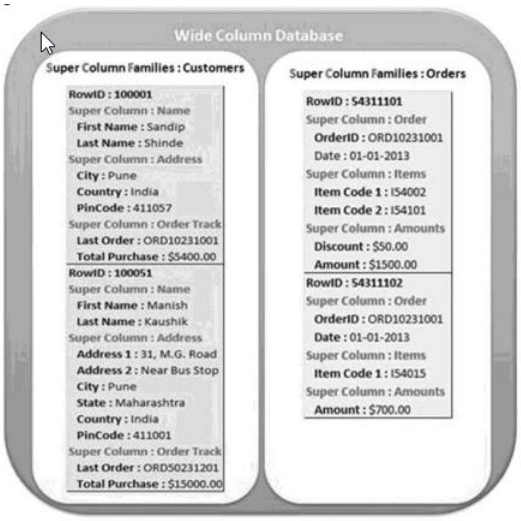

Figura 2. Ejemplo de registros guardados en DB de columnas

Las ventajas principales de guardar datos en columnas sobre los RDBMS yacen en el rápido acceso como búsqueda e inserción de datos, en el caso de las filas estas son guardadas en diferentes lugares de un disco mientras las bases de datos en columnas guardan continuamente los datos en una entrada, facilitando los procesos de búsqueda y acceso.

Ejemplos de bases de datos clave valor pueden ser Bigtable, Hypertable, Cassandra, SimpleDB, DynamoDB

Bases de datos clave valor

Estas bases de datos guardan objetos como identificadores alfanuméricos y valores relacionados, en tablas simples y autónomas (referidas como tablas de hash), los valores pueden ser simplemente strings de texto hasta complejas listas y sets de datos. Las búsquedas de los datos en clave valores solo se pueden ser realizadas hacia las llaves pero no hacia los valores y son limitados a pares exactos [8].

Este tipo de base de datos permite velocidades de consulta mayores que las bases de datos relacionales dándole la utilidad en bases de datos masivas y que requieran alta concurrencia, su simplicidad las hace ideales en el uso de extracción y recuperación de datos y valores necesitados en tareas de aplicaciones como el manejo de perfiles de usuario o sesiones o la recuperación de características de un producto.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Figura 3. Ejemplo de un registro guardado en BD clave valor

Ejemplos de bases de datos clave valor pueden ser Dynamo; Voldemort, Redis, BerkeleyDB, Riak, Cassandra, BigTable o HBase.

Bases de datos documentales

Este tipo de bases de datos se encuentran diseñadas para manejar y guardad documentos, parecidos en estructura al modelo clave-valor, pero su principal diferencia son en los tipos de datos que se guardan los cuales son archivos en formatos XML, JSON (JavaScript Option Notation) o raramente BSON (Binary JSON)[8], otra de las diferencias encontradas con las bases de datos clave-valor es que la columna de valores contiene datos semi estructurados. Una sola columna puede guardar cientos de atributos y los tipos de atributos almacenados pueden variar en gran medida de fila en fila, comparados con el almacenamiento de clave-valor, las llaves y los valores se pueden encontrar en las bases de datos documentadas.

Las Bases de datos documentales tienen una ventaja en el almacenamiento y manejo de colecciones de documentos como textos, mensajes de correo

electrónico e incluso representaciones sin normalizar de una entidad de bases de datos como un producto a un usuario. También son buenos en almacenar datos irregulares en general, de tal manera que con una base de datos relacional, se terminaría poniendo valores nulos si se tratase de identificar en una tabla convencional.

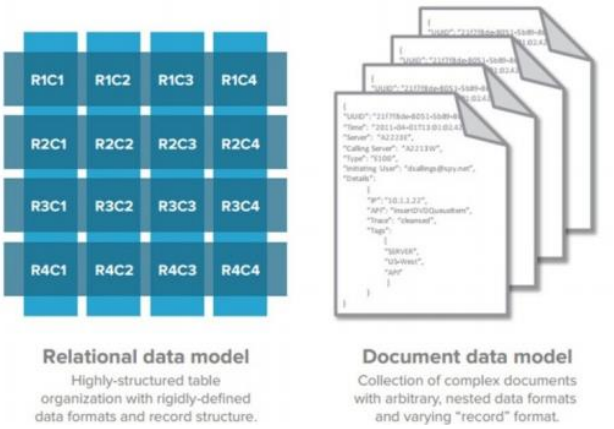

Figura 4. Comparación de bd relacional con bd documental

Algunos ejemplos de este tipo son MongoDB o CouchDB.

Bases de datos orientadas a grafos

Las bases de datos orientadas a grafos remplazan las tablas relacionales con grafos relacionales estructuradas de pares llave-valor. Son similares a las bases de datos orientadas por objetos en el hecho que los grados se representan como una red de nodos orientada en objetos, la información se almacena partiéndola en fragmentos más básicos y estableciendo relaciones entre ellas, cada uno de los nodos tienen relaciones (edges) y propiedades (atributos de objetos expresados como pares clave-valor)[8].

En general las bases de datos basados en grafos son de gran utilidad, en el caso de buscar las relaciones entre los datos que los datos en sí mismos como por ejemplo la representación de redes sociales, investigaciones forenses para la detección de patrones.

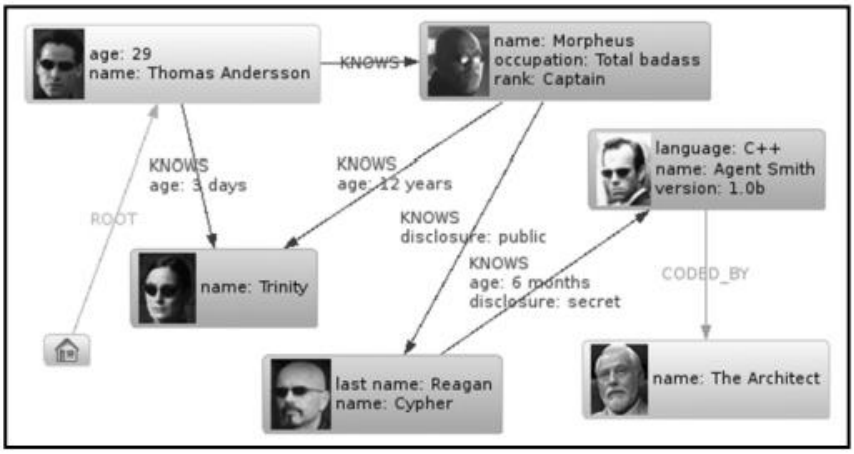

Figura 5. Ejemplo de registros guardados en BD en grafos

Algunos ejemplos de este tipo son Neo4j, InfoGrid Sones GraphDB, AllegroGraph, InfiniteGraph, Virtuoso.

ARQUITECTURA NoSQL MongoDB

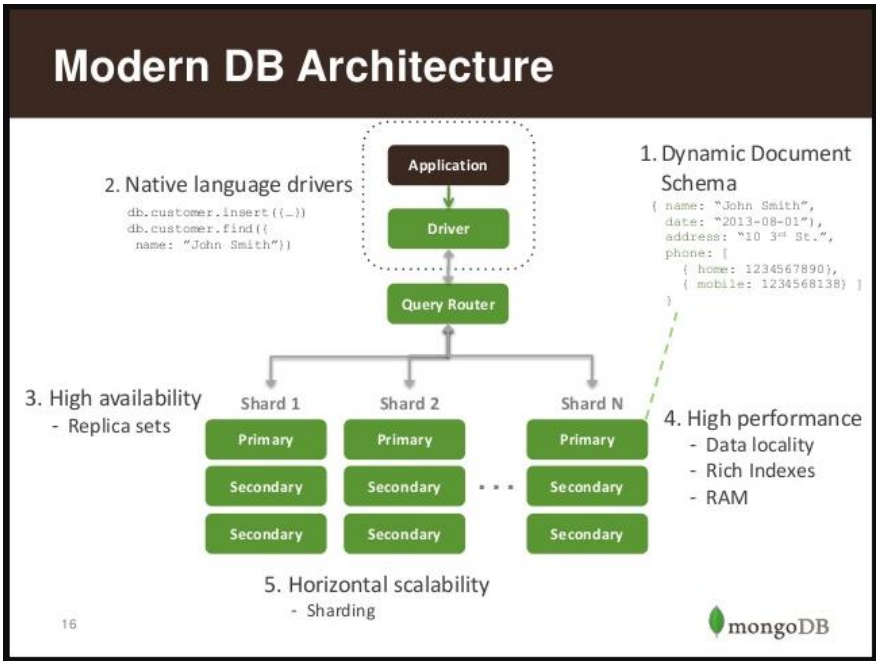

Figura 6. Arquitectura MongoDB

MongoDB es un sistema de base de datos NoSQL en multiplataforma de licencia libre, es una base de datos basado en documentos, lo que implica que cada registro puede tener un esquema de datos distinto, la filosofía de diseño de

MongoDB está enfocada en combinar las capacidades críticas e importantes de las bases de datos relacionales con las innovaciones de las tecnologías NoSQL[9]
.

Es una solución pensada para mejorar la escalabilidad horizontal de la capa de datos con un desarrollo más sencillo; MongoDB cuenta con una serie de herramientas que permiten trabajar con la base de datos desde diferentes perspectivas, y tratar con ella para diferentes propósitos[9].

Mongod: Servidor de bases de datos de MongoDB
Mongo: Cliente para la interacción con la base de datos MongoDB
Mongofiles: Herramienta para trabajar con ficheros directamente sobre la base de datos MongoDB.

La escalabilidad horizontal de la base de datos MongoDB implica trabajar con varios servidores conectados, almacenando en cada uno de los nodos cierta información que debe estar comunicada con el resto de nodos que forman el sistema. Esto dota de mayor flexibilidad al sistema, ya que facilita la agregación como el escalamiento de equipos en función de las necesidades.

MongoDB utiliza el método de Sharding para dividir los datos a lo largo de múltiples servidores de una solución, una tare que también la realizan las bases de datos relacionales[9].

Un shard son uno o varios servidores en un clúster que son responsables de un subconjunto de datos, (un clúster con 1.000.000 de documentos, un shard por ejemplo con 200.000). En el caso de que el shard esté compuesto por más de un servidor, cada uno de estos tendrá una copia idéntica del subconjunto de datos. Para distribuir uniformemente los datos, MongoDB mueve los subconjuntos de shard en shard en base a una clave que debemos elegir. MongoDB también tiene

una herramienta de balanceo, es un proceso de MongoDB para equilibrar los datos en un sistema. Mueve porciones de datos de un shard a otro, de manera automática.

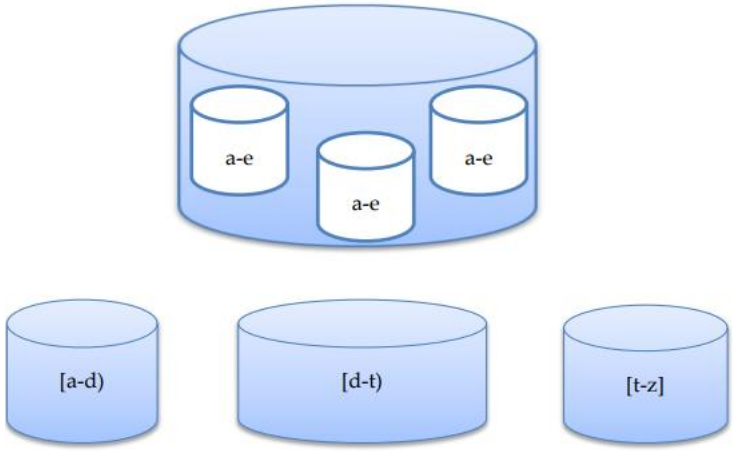

Figura 7. Sharding de MongoDB

MongoDB puede automatizar las consultas para evaluación de la manera más eficiente como sea posible. En la evaluación normalmente incluye la selección datos basados en predicados y el ordenamiento de datos basados en los criterios de clasificación proporcionados.

La consulta de optimización selecciona el mejor índice para usar periódicamente ejecutar planes de consulta alternativos y seleccionar el índice con el mejor tiempo de respuesta para cada tipo de consulta. Los resultados de esta prueba se almacenan como un plan de consulta en caché y se actualizan periódicamente. Los consultores de la base de datos pueden revisar y optimizar los planes utilizando el poderoso método de explicación y filtros de índice.

COMPARACION DE BASE DE DATOS NoSQL

En esta sección vamos a comparar las características de cuatro bases de datos NoSQl con una matriz en base de ciertos atributos, diseño, integridad, indexación, sistema.

NoSQL bases de datos Comparación					
Características		Basado en documentos	Basado en columnas	Llave-Valor	Orientado en Grafos
		MongoDB	Hbase	Redis	Neo4j
Diseño	Data Storage	Volatile memory file system	HDFS	Volatile memory file system	Volatile File System
	Lenguaje de consulta	Volatile memory file system	API calls, XML, Rest,Thrift	API calls	API calls REST SparQL
	Protocolo	Custom binary(BSON)	HTTP/REST Thrift	Telnet-like	HTTP/RESS
	Actualizaciones de entrada condicionales	si	si	no	no
	MapReduce	si	si	no	no
	Unicode	si	si	si	si
	Compresión	si	si	si	
Integridad	modelo de integridad	BASE	Log Replication	no tiene	ASID
	atomicidad	Condicional	si	si	si
	consistencia	si	si	si	si
	isolation	no	no	si	si
	durability	si	si	si	Si
	integridad referencial	no	no	si	Si
Indexación	indexación secundaria	si	si	no	no
	llaves compuestas	si	si	no	no
	búsquedas de textos completa	no	no	no	Si
	soporte de grafos	no	no	no	Si
Sistema	tamaño de los valores	16mb	2TB	no	no
	sistema operativo	Multiplataforma	Multiplataforma	Linux Mac Windows	Multiplataforma
	lenguaje de programación	C++	Java	C, C++	Java

Figura 7. Comparación de BDs respecto a las clases

CONCLUSIONES

La necesidad de las organizaciones de optimizar sus procesos y optimizar sus objetivos organizacionales como estratégicos ha obligado a adquirir herramientas que ayuden a agilizar los procesos de extracción, análisis y consulta de los datos estructurados como no estructurados. Tales requerimientos

computacionales y de almacenamiento para ciertas aplicaciones como las analíticas de Big Data, Inteligencia de Negocios y Social Networking sobre datasets de Petabytes ha empujado a las bases tradicionales en SQL hacia los límites. Esto trajo al desarrollo de bases de datos horizontalmente escalables, distribuidos y no relacionales, propiedades encontradas en las bases de datos NoSQL.

BIBLIOGRAFÍA

[1] Leavitt, N. (2010). Will NoSQL databases live up to their promise?. Computer,43(2), 12-14

[2] Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. IEEE Data Eng. Bull, 32(1), 3-12.

[3] R. Cattell, (2010) "Scalable SQL and NoSQL Data Stores," ACM SIGMODRecord, vol. 39.

[4] I. Pérez, B. León (2007).Lógica difusa para principiantes 1ra ed. Caracas: Texto C.A.

[5] Microsoft SQL Server Databases de la pagina web: http://www.microsoft.com/enus/sqlserver/default.aspx

[6] brewers-cap-theorem-on-distributed-systems de la pagina web: https://www.royans.net/wp/2010/02/14/brewers-cap-theorem-on-distributed-systems/

[7] Henry F. Henry F. Korth, Abraham (2006) “Fundamentos de Bases de Datos”. McGraw-Hill.

[8] Bases de datos NoSQL: arquitectura y ejemplos de aplicación de la pagina web: https://e-archivo.uc3m.es/bitstream/handle/10016/22895/PFC_raul_herranz_gomez_2014.pdf

[9] Mongo DB Achitecture Guide encontrado en la pagina web: https://jira.mongodb.org/secure/attachment/112939/MongoDB_Architecture_Guide.pdf.