

UN VIAJE A TRAVÉS DE BASES DE DATOS ESPACIALES NoSQL

A JOURNEY THROUGH SPATIAL DATABASE NoSQL

RESUMEN

Actualmente, aunque las bases de datos relacionales son utilizadas con éxito para diferentes tipos de escenarios y datos, el concepto NoSQL se escucha con fuerza, a pesar de todas las ventajas que puede ofrecer el modelo relacional, los cuales, a través de su almacén de tuplas y con habilidades de actualización, recuperación de información y estabilidad, pueden complementarse sobre cualquier sistema con gran eficiencia. En este artículo se presenta una revisión global de lo hoy conocido como tendencia NoSQL, analizando sus ventajas en implementaciones de tipo espacial y geográfico. Como objetivo final de este artículo se plantea la identificación de elementos de análisis al momento de desarrollar sistemas espaciales, que le permita al lector identificar si NoSQL es una buena opción como modelo a implementar.

Palabras clave: clave/valor, consistencia, disponibilidad y tolerancia de reparto, nodo, punto de referencia.

ABSTRACT

Currently though relational databases are successfully used for different types of scenarios and data, the concept NoSQL is heard with force, despite all the advantages it can offer the relational model, which, through its stores of tuples and with skills of update, information retrieval, and stability, can be complement on any system with high efficiency. This article shows an overall concept of what is now known as NoSQL trend, analyzing their advantages in implementations of spatial and geographical type. As the ultimate goal of this article arises the identification of elements of analysis at the time of developing space systems, allowing to the reader identify whether NoSQL is a good choice as a model to be implemented.

Key words: availability and delivery tolerance, consistency, key / value, landmark, node.

Helio Henry Ramírez Arévalo

Magister en Educación
Docente Universidad Distrital
Francisco José de Caldas
hhramireza@udistrital.edu.co
Bogotá, Colombia

Jhon Francined Herrera Cubides

Magister en Ingeniería de Sistemas y
Computación
Docente planta de la Universidad
Distrital Francisco José de Caldas
jherrerac@udistrital.edu.co
Bogotá, Colombia

Tipo: Artículo de reflexión

Fecha de Recepción: Octubre 7 de 2013

Fecha de Aceptación: Diciembre 22 de 2013

1. INTRODUCCIÓN

A medida que los requerimientos de usuarios han ido creciendo y especializándose, la información que se gestiona a través los sistemas informáticos que soportan dichos requerimientos ha evolucionado, pasando de cúmulos de información con una estructura clara y definida, y que eventualmente se podían normalizar, escalarlos acorde a su crecimiento y hacerlos persistir desde una capa de aplicación a un repositorio, ya sea de tipo entidad – relación u objeto – relacional [15]; a un cúmulo de información provenientes de diferentes fuentes de datos, con diferentes formatos y estructura, los cuales requieren ser almacenados usando estrategias de persistencia que no cumplen el esquema entidad – relación.

Como respuesta a la demanda de grandes volúmenes de información, dentro de los cuales no se conserva una estructura ni fuentes de datos comunes, tales como la información gestionada por redes sociales (Facebook, Twitter, Google, etc.), buscadores, comunidades y demás componentes soportados sobre ámbitos Web 2.0 y Web 3.0, etc., además del uso de aplicaciones que involucran diferentes capas y componentes de información, como lo son las aplicaciones espaciales; lo que han guiado el mundo de la persistencia y la gestión de datos hacia nuevas arquitecturas de almacenamiento de información, que deben ser de alto rendimiento, escalables y distribuidas, como por ejemplo NoSQL y Big Data [10].

NoSQL (Not Only SQL) corresponde a una estrategia de persistencia que no siguen el modelo de datos relacional, y que no utilizan SQL como lenguaje de consulta; en otras palabras, no están supeditadas a una estructura de datos en forma de tablas y relaciones entre ellas, permitiendo a los usuarios almacenar información en formatos diferentes a los tradicionales, usando estrategias como clave-valor, mapeo de columnas, documentos o grafos.

Como características de las bases de datos NoSQL [4], se identifican la escalabilidad horizontal, la habilidad de distribución, el uso efi-

ciente de los recursos, la libertad de esquema, el uso de modelos de concurrencia débil y las consultas simples.

El presente artículo hace una exploración general sobre los antecedentes de las bases de datos NoSQL, para luego entrar a examinar el concepto que define las bases de datos NoSQL, identificando algunas de las desventajas que presentan dicho tipo de bases de datos; para terminar con una revisión acerca de la incursión de dichas bases de datos en campos profesionales como los ambientes geoespaciales y catástrales.

2. ANTECEDENTES

En el vasto mundo de los sistemas y la informática, la necesidad de hacer persistir la información que soporte el desarrollo de aplicaciones, ha evolucionado hacia escenarios como el uso de métodos orientados a objetos y modelos relacionales para almacenamiento de datos con administración a través de declarativas como de lenguajes de programación, ejecutadas tanto interna como externamente.

Para la implementación de sistemas de información, en muchos escenarios se requiere que la capa de aplicación se comunique y efectúe diferentes transacciones con la capa de persistencia. Al usar lenguajes de programación y lenguajes de base de datos con semánticas y fundamentos diferentes, uno basado en software bajo los principios de ingeniería, y el otro basado netamente en principios matemáticos; se encuentra una "falta de concordancia" o diálogo fluido entre los dos mundos, situación que el desarrollador debe tratar de manejar, presentando al usuario final una interfaz de usuario gráfica donde se manipulen y consulten datos de manera eficiente. Esta unión de esquemas, objetos y bases de datos relacionales lo puede lograr si entiende claramente los paradigmas y diferencias de cada uno.

En primera instancia, es conveniente recordar algunos conceptos acerca de las bases de datos relacionales, como los siguientes:

- Como lo plantea [13], una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional.
- En la terminología formal del modelo relacional, una fila recibe el nombre de tupla, una cabecera de columna es un atributo y el nombre de la tabla una relación. El tipo de dato que describe los valores que pueden aparecer en cada columna está representado por un dominio de posibles valores [7].

Los anteriores conceptos permiten identificar que los datos manejados por un modelo relacional, diseñado como base para un sistema de información de una organización, tiene una estructura definida, que es común y estandarizada, y que debe ser cumplida por todos los inputs que ingresen los procesos o transacciones y los outputs que se generen a la capa de aplicación, y en su articulación con la capa de persistencia.

También es importante contextualizar el significado de una transacción, la cual corresponde a una única operación lógica ("de negocio"). Dichas transacciones para ser confiables deben cumplir con unas propiedades básicas, definidas por Jim Gray en 1970, las cuales fueron agolpadas bajo el acrónimo ACID por Andreas Reuter y Theo Härder en 1983. Estas cuatro propiedades se definen como [22]:

- *Atomicidad*: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Requiere que cada transacción sea "todo o nada": si una parte de la transacción falla, todas las operaciones de la transacción fallan, y por lo tanto la base de datos no sufre cambios.
- *Consistencia*: Es la propiedad que asegura que sólo se empieza aquello que se puede

acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos. La propiedad de consistencia sostiene que cualquier transacción llevará a la base de datos desde un estado válido a otro también válido. Cualquier dato que se escriba en la base de datos tiene que ser válido de acuerdo a todas las reglas definidas, incluyendo (pero no limitado a) los constraints, los cascades, los triggers, y cualquier combinación de estos.

- *Aislamiento* ("Isolation"): es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error. Cada transacción debe ejecutarse en aislamiento total; por ejemplo, si T1 y T2 se ejecutan concurrentemente, luego cada una debe mantenerse independiente de la otra.
- *Durabilidad*: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

Dichas transacciones tiene un comienzo, que se puede especificar en un lenguaje de consulta como SQL, el cual la especifica explícitamente como begin/start transaction. De igual forma, la terminación de una transacción puede darse por cualquiera de las dos siguientes situaciones:

- *Commit*: la transacción actual ha terminado y se ha ejecutado satisfactoriamente.
- *Rollback*: provoca que la transacción se aborte, y regrese a su último estado válido.

Ahora bien, un lenguaje de consulta es un lenguaje en el que un usuario solicita información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consulta pueden clasificarse como [13]:

- *Lenguajes procedimentales*: el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos

para calcular el resultado deseado.

- **Lenguajes no procedimentales:** el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

Dentro de estos lenguajes de consulta se identifica SQL, como uno de los lenguajes comerciales con más amplia aplicación, el cual usa una combinación de álgebra relacional y construcciones del cálculo relacional. Como lo plantea [13], una consulta típica en SQL tiene la siguiente forma, expresada en la ecuación (1).

$$\prod A_1, A_2, \dots, A_n (\sigma_P(r_1 \times r_2 \times \dots \times r_m)) \quad (1)$$

Cada A_i representa un atributo, y cada r_i una relación. P es un predicado.

Select A_1, A_2, \dots, A_n
From r_1, r_2, \dots, r_m
Where P

Lo anteriormente descrito ha venido funcionando a lo largo de la historia, pero en el mundo contemporáneo, la cantidad de información que nos circunda es creciente de forma exponencial. Como lo manifiesta [19], el increíble crecimiento que están experimentando los volúmenes de datos, que según cálculos de IDC, este año sobrepasará los 1,9 billones de gigabytes, nueve veces más rápido que hace cinco años.

Dentro de los factores que influyen en este crecimiento de los volúmenes de datos, [3] identifica el aumento de velocidad de las conexiones a Internet, el de contenidos de usuarios (en las redes sociales, en blogs, en comentarios, etc) o la información geoespacial (datos asociados a localización, gracias al GPS), en términos generales, datos producidos por diferentes fuentes internas y externas, como se muestra en la siguiente figura 1 [23]; el tamaño de los datos asociados a una empresa, una persona, un lugar, una temática, etc.

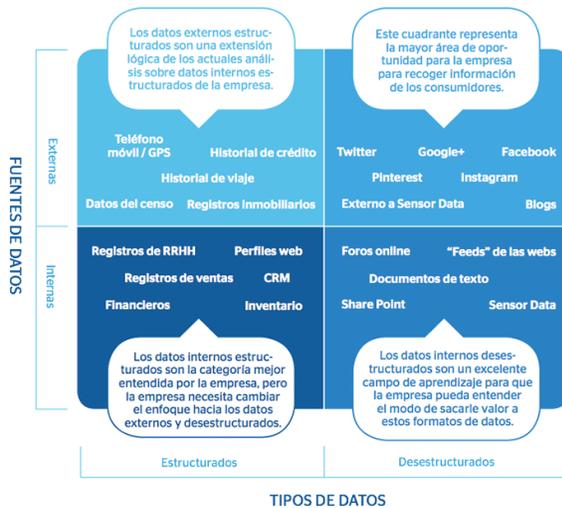


Figura 1. Fuentes y tipos de datos

Bajo este contexto [17] plantea interrogantes como:

- ¿Por qué son tan importantes para las empresas las decisiones tomadas básicamente gracias a los datos estructurados?
- Si el habla y la conversación, juegan un papel fundamental en las interacciones persona a persona, ¿por qué las organizaciones no saben aprovechar todavía la potencialidad que contiene esa información no estructurada presente en miles y miles de conversaciones e interacciones? En el mundo de los negocios, esta información es crítica para mejorar la experiencia del cliente, identificar tendencias emergentes, crear nuevos productos y servicios, y mucho más.

Hoy en día, una gran cantidad de los datos de una empresa son datos no estructurados y tienen un formato de texto, voz o video. Lamentablemente, la mayoría de las empresas no tienen las herramientas necesarias para apreciar la riqueza inmensa de esa información, incluso a la hora de llegar a tomar decisiones vitales para la propia compañía.

De allí surge que se deban procesar datos que no conservan una estructura estándar y en mu-

chos casos con un tamaño excesivamente grande, aspectos que según como lo describe [19], con las bases de datos y arquitecturas tradicionales no se puede procesar, gestionar y analizar tal masa de información, de allí que se estén empezando a demandar soluciones de analítica empresarial que incluyen data mining, estadísticas, optimización y predicciones para convertir de forma proactiva los datos en inteligencia de negocio.

Dichos datos no estructurados pueden no encajar sobre tablas relacionales, provocando irregularidades y problemas de almacenamiento, en especial con contenido controlado por el usuario, por la inexactitud de datos manejados, y como el modelo relacional requiere de representaciones iniciales, complica el ajuste de datos dinámicos.

Estas complicaciones han estimulando la búsqueda de alternativas diferentes sobre las bases de datos relacionales tradicionales, conocidas como almacenes de datos NoSQL, aunque carecen de robustez sobre algunos ambientes y esquemas.

3. CONOCIENDO EL CONCEPTO NoSQL

Para incursionar en el concepto de NoSQL, a continuación se presentan algunos referentes propuestos por [14], donde se describen términos para NoSQL.

El término NoSQL fue acuñado en 1998 por Carlo Strozzi y retomado en 2009 por Eric Evans, quien sugiere mejor referirse a esta familia de bases de datos de nueva generación como “*Big Data*” mientras que Strozzi considera ahora que *NoREL* es un mejor nombre para dicho tipo de bases de datos.

NoSQL – “not only SQL” es una categoría general de sistemas de gestión de bases de datos que difiere de los sistemas manejadores de bases de datos – RDBMS, en aspectos como:

- No tienen schemas, no permiten JOINS, no intentan garantizar ACID y escalan horizontalmente.

- Tanto las bases de datos NoSQL como las relacionales son tipos de almacenamiento estructurado.

La principal diferencia radica en cómo guardan los datos (por ejemplo, almacenamiento de un recibo):

- En una RDBMS, la información a persistir se tendría que partir en diferentes tablas y luego, en la capa de aplicación, usar un lenguaje de programación que realice el respectivo mapeo para transformar estos datos en objetos de la vida real [2].
- En NoSQL, simplemente se guarda el recibo. Dado que NoSQL es libre de schemas, no se debe diseñar tablas y su estructura por adelantado.

De igual forma, [14], identifica algunas características principales, tales como:

- Fáciles de usar en clústeres de balanceo de carga convencionales, debido a que facilitan escalabilidad horizontal.
- Guardan datos persistentes (no sólo cachés).
- No tienen esquemas fijos y permite la migración del esquema sin tener que ser reiniciadas o paradas.
- Suelen tener un sistema de consultas propio en vez de usar un lenguaje de consultas estándar.
- Tienen propiedades ACID en un nodo del clúster y son “eventualmente consistentes” en el clúster.

Por otro lado, como se explicó anteriormente, los sistemas relacionales operan bajo las propiedades ACID definidas para las transacciones. En su defecto, las bases de datos NoSQL son repositorios de almacenamiento más optimistas, y siguen el *modelo BASE* [14]:

- *Basic availability*: el almacén funciona la mayoría del tiempo incluso ante fallos gracias al almacenamiento distribuido y replicado.
- *Soft-sate*: los almacenes no tienen porque ser consistentes ni sus réplicas en todo mo-

mento.

- *Eventual consistency*: la consistencia se da eventualmente.

Como lo describe [2], un primer intento para conceptualizar el término NoSQL se evidenció en la publicación “*NoSQL: Einstieg in die Welt nichtrelationaler Web-2.0-Datenbanken*” de S. Edlich et al. (2010), donde se definió como una nueva generación de sistemas de bases de datos que tenía al menos una de las siguientes propiedades:

- El modelo de datos subyacente no es relacional.
- Los sistemas están diseñados desde el principio como escalable horizontalmente y verticalmente.
- El sistema es de código abierto.
- El sistema es esquema-libre o sólo tiene restricciones de esquema suaves.
- Debido a la arquitectura distribuida, el sistema admite un método de replicación de datos simple.
- El sistema proporciona una API sencilla.
- El sistema generalmente utiliza un modelo de consistencia diferente.

El término NoSQL se debe enfocar como un conjunto de características para diferentes almacenes de datos, sin un dominio homogéneo. Actualmente existen diferentes categorías de frameworks NoSQL [2], dentro de los cuales se identifican en la figura 2.

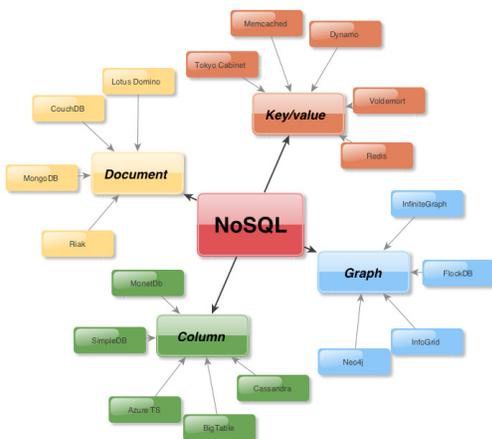


Figura 2. Familia NoSQL

Llave/ Valor: almacena datos en pares llave / valor. Son muy eficientes por su desempeño y alta escalabilidad, pero son difíciles de consultar y de implementar problemas del mundo real [21]. En la tabla 1 se presenta un ejemplo de par llave / valor.

Tabla 1. Ejemplo de Par Llave / Valor
Fuente: [21]

Key	Value
F911c336e5b89e	Content-Type: image/gif ROIgODlhKwIlAXcAMSH+GINvZnR 3YXJlOiBNaWZ AhsBgQAAAAAAP8AAPHI+py+0P o5y0AiEM3pl1e

Columnas: almacenan datos en estructuras tabulares, pero las columnas pueden variar en el tiempo y cada fila puede tener solamente un subconjunto de columnas.

Orientado a documentos: tal como llave / valor, pero permiten almacenar mas valores por cada llave. Un valor tipo documento podría ser por ejemplo un XML o un fragmento JSON. Este es un buen paradigma para programadores dado su facilidad, especialmente por el uso de lenguajes de script, implementa mapeo uno a uno para relaciones entre los objetos de código y los objetos (documentos) en el sistema.

Grafos: almacena objetos y relaciones en nodos y vértices de un grafo. Para situaciones que se ajusten a este modelo, como datos jerárquicos, esta solución podría ser más rápida que las otras.

En la actualidad las bases de datos NoSQL se emplean para el almacenamiento de datos que manejan un gran volumen, no estructurados, sobre aplicaciones web, y ya se encuentran implementaciones en sistemas de búsquedas como Amazon, Google, Yahoo o Facebook.

4. TRABAJANDO CON NoSQL EN SISTEMAS GEOESPACIALES

Como se explico anteriormente, las bases de datos NoSQL están diseñados para grandes volúmenes de datos y aplicaciones Web 2.0. En el

contexto de la Web 2.0 la dimensión espacial es de interés particular, para un sinnúmero de aplicaciones como Flickr o Panoramio que permiten al usuario georreferenciar su contenido. Algunos intentos hacia el almacenamiento de datos espaciales en bases de datos de documentos se han llevado a cabo, como por ejemplo MongoDB y CouchDB. Además, la especificación de GeoJSON demuestra que las capacidades espaciales de los almacenes basados en documentos serán ampliadas. Además, las bases de datos basadas en grafos admiten datos espaciales de forma nativa [18].

Por ende, al momento de determinar cual modelo de base de datos seleccionar para aplicaciones de tipo espacial, se debe tener en cuenta aspectos como velocidad de procesamiento para consulta sobre la base de datos, requisitos de espacio, escalabilidad, facilidad de manejo y estabilidad; y basarse en la documentación y experiencia que se pueda encontrar.

Los sistemas que manejan datos geográficos, deben representar de una forma simple la mayor cantidad de aspectos del mundo real, ya que la representación exacta de todas las características de un paisaje es imposible de representar en una computadora, por lo subjetivo de la descripción de ubicación y categorización de elementos como carreteras, árboles, campos abiertos, ríos o edificaciones, junto con las referencias geográficas que se puedan utilizar. Dependiendo de su finalidad los datos pueden ser simples o demasiado complejos.

Dado las especificidades, tipos y volúmenes de datos que manejan este tipo de sistemas, se requiere de un proceso en equipos de cómputo interconectados, ya sea sobre tecnología de red convencional o Internet, en términos generales, aprovechando las ventajas del procesamiento distribuido, enmarcado en la aplicación del precepto de la teoría general de sistemas: “el todo es la suma de sus partes”. Este requerimiento surge de la necesidad de acceder y usar la data espacial para ubicaciones particulares, almacenada en diferentes locaciones físicas y a menudo con diferentes estándares y formatos [20].

Para ello, y teniendo en cuenta el teorema CAP, sobre el cual muchos sistemas NoSQL están diseñados [2], se deben tener en cuenta tres requerimientos especiales durante el diseño del ambiente distribuido:

- *Consistencia*: todos los nodos ven los mismos datos al mismo tiempo.
- *Disponibilidad (Availability)*: garantiza que cada petición recibe una respuesta acerca de si tuvo éxito o no. La disponibilidad está pensada en que el modelo soporte todos los tipos de datos y transacciones disponibles, y así un nodo falle los otros nodos puedan continuar sin ningún problema.
- *Tolerancia a la partición (Partition)*: el sistema continúa funcionando a pesar de la pérdida de mensajes. En cuanto a la tolerancia de particiones, esta debe permitir que interactúe entre componentes individuales distribuidos en todo el ambiente, pensando que el sistema continúe cuando algunos componentes se pierden.

Estas tres características no necesariamente deben ser aplicadas como un todo y en igual proporciones, ya que para diferentes sistemas pueden variar los niveles de cada una, dependiendo del objetivo para el cual fueron diseñados. A continuación, como se muestra en figura 3, se presenta las diferentes bases de datos NoSQL según su aplicación en el teorema de CAP [21].

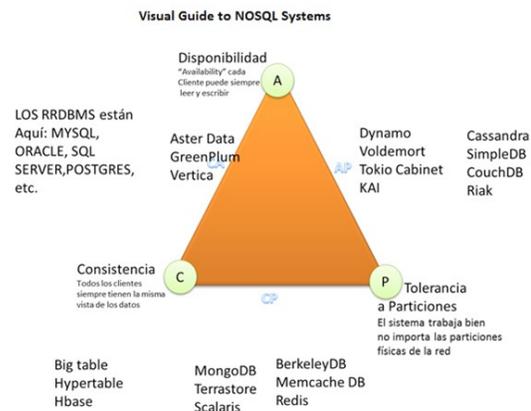


Figura 3. Teorema de CAP con una visión general de las bases de datos

Dado las características y los tipos de datos que manejan los sistemas geoespaciales, es importante que el diseño de los ambientes distribuidos que soportan dichos sistemas tengan en cuenta las características citadas anteriormente. Una vez se consolide el ambiente distribuido que soporta el procesamiento del sistema geoespacial, se debe verificar elementos como el modelamiento de datos, los tipos de datos geográficos usados, el soporte de sistemas de referencia espacial, entre otras elementos, que soportaran las funcionalidades disponibles para el apoyo geodésico, tales como map / reduce, generación de índices espaciales, procesamiento de consultas, predicción de desastres naturales, etc. [6].

Algunas soluciones NoSQL que soportan datos geográficos pueden ser nativas o con extensiones; otros, aunque inicialmente no están diseñados para soportar datos geográficos, pueden ser mejorados para soportarlos. Dentro de los proyectos que soportan datos espaciales se encuentran:

- *Neo4j*: es un almacén de datos NoSQL de entorno gráficos utilizado en la consulta de datos espaciales vectoriales con integración gráfica que interactúa con aplicaciones Java o cualquier otra biblioteca, y que muestra sus datos como un gráfico. Al ser gráfico, los beneficios sobre consultas son mayores sobre recorridos o regiones locales, análisis trayectorias cortas y consultas de conectividad entre ubicaciones [2].

Neo4j maneja licencia GPLv3, es escrito en Java y se centra en el rendimiento grafos/ relaciones, como se muestra en la figura 4. Para la ejecución de consultas hace uso del protocolo HTTP/REST o directamente sobre Java. Brinda soporte a datos espaciales a través de un plugin llamada Neo4j-Espacial, que soporta geometría, puntos, cadenas de líneas, polígonos, multipuntos, funciones simples de muti cadenas de líneas y multi polígonos. Hace uso de R-Tree para consultas espaciales y maneja un modelo de datos único, almacenamiento de objetos y relaciones de nodos y aristas en un grafo.

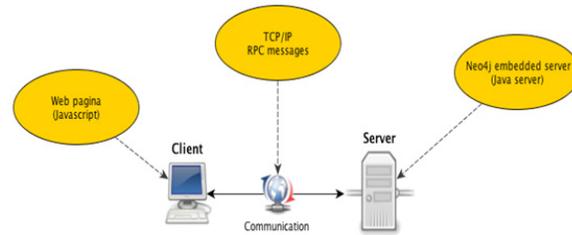


Figura 4. Arquitectura Neo4j – espacial

Esta base de datos es usada por compañías como Adobe, Cisco o Squidoo [11].

- *CouchDB*: es una base de datos NoSQL que emplea JSON para almacenar los datos, como se muestra en la figura 5; JavaScript como lenguaje de consulta por medio de Map/Reduce y HTTP como API [21]. Una de sus características más peculiares es la facilidad con la que permite hacer replications. Es un almacén de datos orientado a documentos licenciado por Apache y escrito en Erlang, su objetivo es la consistencia de datos y facilidad de uso. CouchDB maneja una extensión espacial llamada GeoCouch que soporta datos espaciales, con B-tree para búsquedas de clave/valor y R-tree para consultas espaciales [1].

```
{
  "_id": "a26ab886a7d200a93a32ae192200416d",
  "_rev": "4-c97d88df8c79468bcbe03fcf0e5921d",
  "another_attribute": 17,
  "yet_another_attribute": "a_string",
  "_attachments": {
    "Screenshot.png": {
      "content_type": "image/png",
      "revpos": 3,
      "length": 22666,
      "stub": true
    }
  },
  "trace.nam": {
    "content_type": "application/octet-stream",
    "revpos": 2,
    "length": 7830,
    "stub": true
  }
}
```

Figura 5. Ejemplo de un documento JSON en CouchDB

CouchDB combina un modelo intuitivo de almacenamiento de documentos con un poderoso motor de consultas [5]. En este sistema se manejan normalmente geometría tales como puntos, líneas y polígonos, con consultas limitadas en áreas, polígono

y con radio de búsquedas. Utiliza Map/Reduce para realizar consultas. Aunque es de uso sencillo su rendimiento es bajo cuando el conjunto de datos espaciales es muy grande.

A diferencia de una base de datos relacional, CouchDB no almacena los datos y sus relaciones en tablas. En cambio, cada base de datos es una colección de documentos independientes. Cada documento mantiene sus propios datos y su esquema auto contenido. Una aplicación puede acceder a múltiples bases de datos, por ejemplo una residente en el teléfono móvil del usuario y otra residente en un servidor. Los metadatos del documento contienen información acerca de la versión del mismo, permitiendo refundir cualesquiera diferencias que puedan haberse producido mientras las bases de datos estaban desconectadas [24].

CouchDB es usado a gran escala por compañías como Ubuntu, la BBC o Credit Suisse, es ideal para aplicaciones móviles debido a su capacidad de replicación y sincronización, y para CRM o CMS en el que sea importante disponer de los cambios de versiones de un documento [11].

- *MongoDB*: es también un almacén de datos con orientación por documento, con licencia AGPL. Es reconocida como la base de datos NoSQL líder, con una mayor productividad y una mejor experiencia de desarrollo. Sus características incluyen un modelo de datos JSON con esquemas dinámicos, amplia compatibilidad de controladores, auto-sharding, replicación integrada y de alta disponibilidad, soporte índice completo y flexible, consultas enriquecidas, agregación, actualizaciones en contexto y GridFS para el almacenamiento de archivos de gran tamaño [16].

Como se muestra en la figura 6, MongoDB es una base de datos ágil que permite a los esquemas cambiar rápidamente tal como evolucionan las aplicaciones, sin dejar de ofrecer las funcionalidades que los desa-

rolladores esperan de las bases de datos tradicionales, como los índices secundarios, un lenguaje de consulta plena y estricta consistencia.

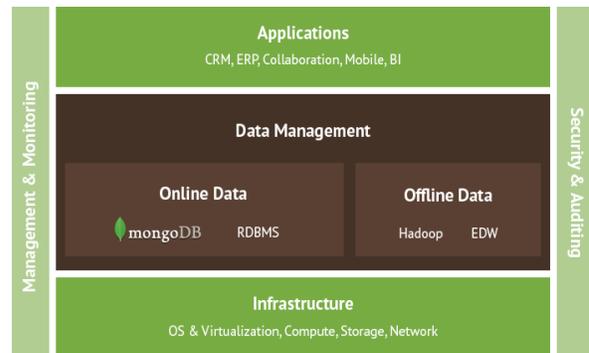


Figura 6. Articulación MongoDB

MongoDB es usada en aplicaciones famosas como Foursquare, Sourceforge, Google, Facebook o New York Times [11].

En términos generales, Neo4j tiene un desempeño más rápido en relación con estos otros dos sistemas, independientemente del tamaño de los datos, con opciones de búsquedas de proximidad y análisis de ruta, pero si se desea aprovechar todas las características espaciales, se necesita el uso de Java directamente sobre el código fuente, ya que si se maneja con interfaz HTTP solo se pueden hacer consultas parciales.

Otras bases de datos NoSQL son [11]:

- *Cassandra*: desarrollado originalmente por Facebook para su sistema de mensajería posteriormente donado a la Apache Software Foundation, es una base de datos basada en el modelo clave-valor, su arquitectura está basada en una serie de nodos iguales que se comunican con un protocolo P2P con lo que la redundancia es máxima. En las últimas versiones han implementado un lenguaje de consulta llamado CQL (Cassandra Query Language) tratando de asemejarlo con SQL. Cassandra es usada en grandes aplicaciones como Twitter, YouTube o Netflix.
- *Riak*: Base de datos NoSQL escrita en Erlang, basada en el documento escrito por

Amazon sobre el sistema de almacenamiento Dynamo, utilizando el modelo de clave-valor para el almacenamiento de datos. Dispone de una Open Source (que es completa y funcional) y otra Enterprise (que agrega facilidades como por ejemplo como una interfaz de administración). Este motor de base de datos es utilizado por múltiples compañías como Best Buy, AOL, Ask.com.

- *Voldemort*: Es el sistema de base de datos basado en el modelo clave-valor de LinkedIn, escrito en Java, fue creado para solucionar un problema de escalabilidad que tenía con las base de datos relacionales, cuenta con una comunidad activa desde su liberación.

5. DESVENTAJAS DE NoSQL

NoSQL no es ajena a presentar determinadas desventajas, dentro de las cuales se identifican [9]:

- *Falta de estándares*: en NoSQL como Open-Source, muchas veces el desarrollador se convierte en el “optimizador de las consultas de datos”, lo que sería un retroceso de 20 o 30 años en la manipulación de datos a gran escala. Al tratarse de una tecnología emergente donde no hay estándares y las alternativas son múltiples, existe un grado importante de no compatibilidad entre herramientas; cada base de datos NoSQL tiene sus propias interfaces, APIs, etc. Por otra parte, tampoco existe un soporte tan robusto, lo que podría mejorar al ir madurando la tecnología y decantando en sólo algunas opciones más usadas.
- *Falta de rigurosidad y documentación*: “muchos de estos sistemas han sido desarrollados por hackers que poco interés tienen en entender por qué los sistemas funcionan y por lo tanto es muy difícil evaluar de manera rigurosa los pro y los contra de ellos”.
- *Punto débil*: (fiabilidad) mientras ofrecen la gestión de un mayor volumen y variedad de datos con mayor agilidad, no necesariamente brindan garantías de fiabilidad de la data. Puntualmente, “su mayor desventaja

se presenta en que al priorizar por la disponibilidad y la inmediatez del dato para el usuario, trabajan con datos distribuidos y a veces replicados”. Si el dato es muy volátil, puede generar inconsistencia o distintas versiones, algo que puede complicar a las aplicaciones transaccionales.

6. USO DE BASE DE DATOS NoSQL EN SISTEMAS CATASTRALES

Como lo expone [18], un tema interesante de amplia aplicación es el manejo de documentos catastrales, con por ejemplo el manejo de registros o contratos catastrales de las propiedades. Estos registros manejan datos variables como tipo de propiedad, número de registro, descripción de la propiedad, situación legal, infraestructura, precio de compra, defectos de responsabilidad, entre otros. Algunos de estos registros están descritos de manera no estructurada y no forman parte de un contrato por sí mismos y / o pueden variar en su contenido. Los elementos del contrato que difícilmente se describen utilizando un esquema predefinido pueden ser vistos como datos no estructurados, ya que no tienen esquema común.

Esta base de datos es dinámica, con opción de consulta y actualización constante. Estos sistemas de manejo de documentos catastrales se implementarían sobre temas de ordenamiento territorial, dominio espacial y en general sobre cualquier aplicación con manejo de documentos de tipo catastral.

La minería de datos en el manejo de documentos puede optimizarse al agregar dimensiones y relaciones espaciales, que apoyan la indexación espacial y el almacenamiento de datos espaciales, permitiendo así que las consultas espaciales que abordan cuestiones de ordenamiento territorial pueden ser resueltas usando minería de datos no estructurados.

De igual forma, como lo expone [8], la mayoría de los motores de bases de datos usados para la gestión territorial, son sistemas de bases de datos geográficas y relacionales. Pero para la gestión de grandes volúmenes de datos y, sobre

todo, si la finalidad es que pueda ser un servicio web, cabe destacar los sistemas de bases de datos NoSQL. Estos permiten un manejo rápido de la información almacenada, puesto que no debe hacer una comprobación registro a registro, reducen al mínimo los accesos a disco, y pueden almacenar cantidades enormes de datos y a la vez dar servicio a multitud de peticiones. Este tipo de bases de datos se han popularizado gracias sobre todo a las modernas redes sociales, y al fin y al cabo, se pretende un modelo de relaciones entre unas parcelas y otras debido a las continuas actualizaciones. En este sentido, cada parcela tiene una relación con otra a partir de actualizaciones topológicas de su geometría, lo que nos lleva a una red parcelaria con conexiones entre ellas.

El modelo más difundido esta comúnmente asociado a las bases de datos relacionales, y aunque este tipo de sistema podrá dar solución a este tipo de modelos parcelarios, estudios anteriores muestran que no ofrece la sencillez que se busca. Las bases de datos NoSQL ofrecen un uso de los recursos más reducido que las bases de datos relacionales y un incremento de la velocidad conforme aumentan los registros.

Estos sistemas de manejo de bases de datos de tipo NoSQL, son populares también en aplicaciones espaciales, dispositivos de apoyo georeferenciada con recopilación de datos, teléfonos inteligentes e implementaciones en Sistemas de Posición Geográfico. El integrar datos de tipo geográficos que están almacenados en diferentes fuentes de información, es un desafío en sistemas geoespaciales [12]. Para ello, los datos deben estar contenidos en objetos para que se asocien con la infraestructura implementada.

Lo interesante de las Bases de Datos NoSQL es

que al ser un término relativamente nuevo, con gran potencial sobre sistemas de tipo geográfico y con todo lo que se puede explotar e implementar en el, las posibilidades de investigación y aplicación en este ambiente son numerosas.

7. CONCLUSIÓN

En términos generales, las bases de datos NOSQL se caracterizan por ser bases de datos que no se rigen por las características y modelos de las bases de datos relacionales. Dentro de las diferencias con los modelos relacionales, entre otras se tiene que no tienen esquemas, no usan SQL, no garantizan la propiedad ACID, en su lugar usan el modelo BASE; abordan el procesamiento de altos volúmenes de información junto con la transaccionalidad que ello implica.

Aunque muchas implementaciones actuales están desarrolladas bajo modelos con bases de datos no relacionales, con el constante trabajo bajo sistemas de tipo distribuido, aplicaciones web y utilización de datos no estructurados, sobre lo cual está diseñado el modelo NoSQL, aumenta la probabilidad del uso de éste con mas auge en el desarrollo de nuevos sistemas, y en específico en sistema de tipo geoespacial, por el tipo de información y estructura que estos manejan.

Por ser una tecnología relativamente emergente, afronta problemas como la falta de estandarización, documentación y fiabilidad, pero de igual forma, su incursión en campos donde se requiere por ejemplo el manejo de datos no estructurados, como los sistemas espaciales; va abriendo camino para fortalecer dichas desventajas, mejorar sus capacidades actuales e implementar nuevas funcionalidades que le permitan alcanzar su madurez.

Referencias Bibliográficas

- [1] J. Anderson, J. Lehnardt, N. Slater; CouchDB: The Definitive Guide. O'Reilly Media, Inc., 2010. [En Línea], consultado el 1 de Noviembre de 2013, disponible en <http://www.hashdoc.com/document/10309/couchdb-the-definitive->

- guide..
- [2] B. Baas; NoSQL spatial: Neo4j versus PostGIS. GIMA - Geographical Information Management And Applications, May 22, 2012. . [En Línea], consultado el de Noviembre de 2013, disponible en <http://repository.tudelft.nl/view/ir/uid:a47d3b8e-650a-4152-a310-366db0773848/>.
- [3] Bitext.com; El boom de la gestión de datos no estructurados. [En Línea], consultado el 20 de Noviembre de 2013., disponible en: <http://www.bitext.com/es/2011/10/el-boom-de-la-gestion-de-datos-no-estructurados.html>.
- [4] A. Castro, J. González, M. Callejas; Utilidad y Funcionamiento de las Bases de Datos NoSQL. [En Línea], consultado el 10 de Noviembre de 2013, disponible en: <http://virtual.uptc.edu.co/revistas2013f/index.php/ingenieria/article/view/2115/2078>.
- [5] Couch DB Relax; [En Línea], consultado el 25 de Noviembre de 2013, disponible en <http://docs.couchdb.org/en/latest/intro/why.html>.
- [6] C. De Souza Baptista, O. Guimarães de Oliveira, F. Gomes de Andrade, T. Da Silva, C. Santos Pires; Using GC Services to Interoperate Spatial Data Stored in SQL and NoSQL Databases. Laboratory of Information Systems, Computer Science Department Federal University of Campina Grande (UFCG). Brazil, 2011.
- [7] R. Elmasri, S. Navathe; Fundamentos de Sistemas de Bases De datos. Pearson. 2007.
- [8] M. Fernández Moreno, A. Morte; Modelo de datos para la monitorización de cambios geométricos del parcelario catastral. Computer Enviroment and Urban Systems. 2013. [En Línea], consultado el 15 de Noviembre de 2013, disponible en <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18857/6/mfernandezmorenoTFM0113memoria.pdf>.
- [9] Gerencia; NoSQL ¿El fin del reinado de las bases de datos relacionales?. Abril de 2013. [En Línea], consultado el 20 de Noviembre de 2013, disponible en: <http://www.emb.cl/gerencia/articulo.mvc?xid=105>.
- [10] L. Glez; Nuevos retos en la gestión de datos: NoSQL y Big Data. Laboratorio de Bases de Datos, Universidad de la Coruña. 2013. [En Línea], consultado el 10 de noviembre de 2013, disponible en http://coba.dc.fi.udc.es/~lgares/FIC_OnDev/NoSQL_BigData_LuisAres.pdf.
- [11] HispaBigData; Base de Datos NoSQL. [En Línea], consultada el 24 de Noviembre de 2013, disponible en <http://www.hispabigdata.es/2013/08/bases-de-datos-nosql.html>.
- [12] Ingeniería y Arquitectura; Cartografía Raster Vectorial adaptada en entorno SIG; [En Línea], consultada el 3 de diciembre de 2012, disponible en: http://www.ingenieriayarquitectura.com/ingenieria_Cartografia_raster_vectorial_adapatada_a_entorno_SIG.html.
- [13] A. Silberschatz, H. Korth, S. Sudarshan; Fundamentos de Bases de Datos. McGRAW-HILL. 2002.
- [14] D. López-de-Ipiña; NoSQL: Introducción a las Bases de Datos no estructuradas. Curso de Verano Big Data & Data Science, Universidad de Santiago de Compostela. 2013. [En Línea], consultado el 18 de noviembre de 2013, disponible en <http://eventos.citius.usc.es/bigdata/>.
- [15] J. Mármol Castillo; Persistencia de Objetos JDO, Solución Java. Facultad de Informática, Universidad de Murcia. [En Línea], consultado el 15 de noviembre de 2013, disponible en: <http://dis.um.es/~jmolina/Persistencia%20de%20Objeto%20JDO.pdf>.
- [16] Mongo DB. [En Línea], consultado el 15 de Noviembre de 2013, disponible en: <http://www.mongodb.org/>.
- [17] D. Parcell; Datos no estructurados: más allá del CRM. [En Línea], consultado el 22 de Noviembre de 2013, disponible en: <http://www.puromarketing.com/53/11699/datos-estructurados-alla.html>. 2011.
- [18] J. Scholz; Coping with Dynamic, Unstructured Data Sets – NoSQL a

- Buzzword or a Savior?. TU Vienna, Gusshausstr. REAL CORP Tagungsband, ISBN: 978-3-9503110-0-6 (CD-ROM); ISBN: 978-3-9503110-1-3 (Print). Mayo de 2011. [En Línea], consultado el 6 de Noviembre de 2013, disponible en http://www.corp.at/archive/CORP2011_76.pdf.
- [19] L. Serrano; La gestión de los datos no estructurados se convierte en la gallina de los huevos de oro. [En Línea], consultado el 23 de Noviembre de 2013, disponible en: <http://www.computing.es/comunicaciones/noticias/1036522000301/gestion-datos-no-estructurados.1.html>.
- [20] S. Steiniger, A. Hunter; Free and Open Source GIS Software for Building a Spatial Data Infrastructure. Department of Geomatics Engineering, University of Calgary. University Drive NW, Calgary. 2012.
- [21] S. Weber; NoSQL Databases. Master of Science in Engineering, University of Applied Sciences HTW Chur, Switzerland. [En Línea], consultado el 6 de Noviembre de 2013, disponible en http://wiki.hsr.ch/Datenbanken/files/Weber_NoSQL_Paper.pdf.
- [22] Wikipedia; Propiedades ACID. [En Línea], consultado el 8 de Noviembre de 2013, disponible en <http://es.wikipedia.org/wiki/ACID>.
- [23] BBVA; Big Data, En qué punto estamos?. [En Línea], consultado el 8 de Noviembre de 2013, disponible en <https://www.centrodeinnovacionbbva.com/magazines/innovation-edge/publications/21-big-data/posts/153-big-data-en-que-punto-estamos>.
- [24] Wikipedia; CouchDB. [En Línea], consultado el 10 de Noviembre de 2013, disponible en <http://es.wikipedia.org/wiki/CouchDB>.