



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA CREAR SITIOS WEB DE ALTO RENDIMIENTO

Sergio Eduardo Mancilla Escobar

Asesorado por el Ing. Sergio Arnaldo Méndez Aguilar

Guatemala, agosto de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA
CREAR SITIOS WEB DE ALTO RENDIMIENTO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

SERGIO EDUARDO MANCILLA ESCOBAR

ASESORADO POR EL ING. SERGIO ARNALDO MÉNDEZ AGUILAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, AGOSTO DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|-------------------------------------|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| VOCAL I | Ing. Alfredo Enrique Beber Aceituno |
| VOCAL II | Ing. Pedro Antonio Aguilar Polanco |
| VOCAL III | Inga. Elvia Miriam Ruballos Samayoa |
| VOCAL IV | Br. Walter Rafael Véliz Muñoz |
| VOCAL V | Br. Sergio Alejandro Donis Soto |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

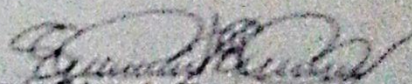
| | |
|-------------|-------------------------------------|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| EXAMINADOR | Ing. Marlon Antonio Pérez Turk |
| EXAMINADORA | Inga. Sonia Yolanda Castañeda |
| EXAMINADORA | Inga. Floriza Felipa Ávila Pesquera |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA
CREAR SITIOS WEB DE ALTO RENDIMIENTO**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha junio de 2012.


Sergio Eduardo Mancilla Escobar

Guatemala, 26 de febrero de 2013

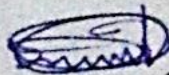
Ingeniero
Carlos Azurdía
Escuela de Ciencias y Sistemas
Facultad de Ingeniería

Respetable Ingeniero Azurdía:

Por este medio le informo, que como asesor del trabajo de graduación del estudiante universitario de la carrera de Ingeniería en Ciencias y Sistemas, **SERGIO EDUARDO MANCILLA ESCOBAR**, carné **200714244**, hago constar que ha finalizado todos los capítulos del trabajo de investigación titulado: **USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA CREAR SITIOS WEB DE ALTO RENDIMIENTO**, el cuál he tenido la oportunidad de revisar y doy mi aprobación al mismo.

Agradeciendo su atención a la presente,

Atentamente,



Sergio Arnaldo Méndez Aguilar
Ingeniero en Ciencias y Sistemas
Colegiado No. 10958
Ing. Sergio Méndez

Catedrático Sistemas Operativos 1
Asesor de trabajo de graduación
Colegiado: 10958



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 06 de Marzo de 2013

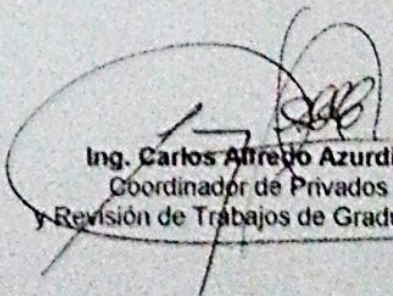
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

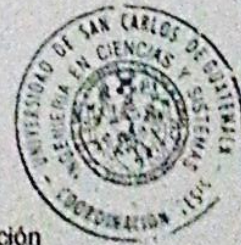
Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **SERGIO EDUARDO MANCILLA ESCOBAR** carnè **200714244**, titulado: **"USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA CREAR SITIOS WEB DE ALTO RENDIMIENTO"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL.: 24767644

E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA CREAR SITIOS WEB DE ALTO RENDIMIENTO”**, realizado por el estudiante **SERGIO EDUARDO MANCILLA ESCOBAR**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”



Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 02 de agosto 2013

ACTO QUE DEDICO A:

- Dios** Por haberme permitido llegar hasta este punto importante en mi vida, gracias por su infinito amor y bondad.
- Mi familia** Por su apoyo incondicional a lo largo de estos años, sin ellos esta meta no hubiera sido alcanzada.
- Ing. Sergio Méndez** Por sus valiosos consejos brindados, que permiten hoy completar esta meta. Gracias.
- Mis amigos** Jerry Osorio, Ángel Navarro, Luis Zambrano y Óscar Porres y Sonia Guamuch, los cuales fueron un apoyo y a quienes considero como mis hermanos.

AGRADECIMIENTOS A:

**La Universidad de San
Carlos de Guatemala**

Por ser mi alma máter de estudios la cual me enseñó lo académico y experiencias de vida.

Mi padre

Rolando por haberme apoyado en todo momento, las desveladas junto a mí en las arduas noches de proyectos, por cuidarme y su amor incondicional.

Mi madre

Silvia por el apoyo incondicional, los consejos invaluable, por la motivación y todo su amor

Mis familiares

Especialmente a mi hermano Byron, por los consejos académicos; a mi cuñada Karina Arenas, por estar conmigo en los momentos difíciles.

| | | |
|--------|--|----|
| 1.5.2. | Manipulación de información..... | 12 |
| 1.5.3. | Fiabilidad de transacciones..... | 12 |
| 1.5.4. | Escalabilidad y elasticidad a través de técnicas de particionamiento | 13 |
| 1.5.5. | Alta disponibilidad | 14 |
| 1.6. | Ventajas en ambientes de producción utilizando bases de datos NoSQL documentales | 14 |
| 1.7. | MongoDB como base de datos NoSQL documental..... | 15 |
| 2. | FACTORES A CONSIDERAR PARA IMPLEMENTAR UNA BASE DE DATOS NOSQL DOCUMENTAL | 17 |
| 2.1. | Grado de adaptación..... | 17 |
| 2.1.1. | Tiempo medio de aceptación y adaptación | 17 |
| 2.1.2. | Nivel de curva de aprendizaje | 19 |
| 2.2. | Grado de facilidad de implementación | 19 |
| 2.2.1. | Requisitos mínimos de implementación | 19 |
| 2.3. | Portabilidad respecto a la migración de datos..... | 20 |
| 2.4. | Casos de éxito de bases de datos NoSQL documentales | 22 |
| 3. | COSTOS DE IMPLEMENTACIÓN..... | 25 |
| 3.1. | Costos de aplicación | 25 |
| 3.1.1. | Licenciamiento | 25 |
| 3.1.2. | Capacitación..... | 27 |
| 3.1.3. | Soporte..... | 27 |
| 3.2. | Costos de infraestructura | 29 |
| 4. | PRUEBAS DE RENDIMIENTO BASE DE DATOS NOSQL DOCUMENTAL VERSUS SQL RELACIONAL | 31 |
| 4.1. | Definición de ambiente de prueba..... | 31 |

| | | |
|----------------------|--|----|
| 4.1.1. | Definición de escenarios de prueba..... | 34 |
| 4.1.2. | VARIABLES A EVALUAR | 35 |
| 4.2. | ANÁLISIS COMPARATIVO PostgreSQL, MySQL y MongoDB | |
| | 10gen..... | 36 |
| 4.2.1. | Tiempos de respuesta | 36 |
| 4.2.2. | Consumo de recursos..... | 38 |
| 4.2.3. | Respuesta a fallos | 46 |
| 4.2.4. | Disponibilidad | 50 |
| 4.3. | Evaluación de resultados..... | 53 |
| CONCLUSIONES | | 57 |
| RECOMENDACIONES..... | | 59 |
| BIBLIOGRAFÍA..... | | 61 |
| ANEXOS..... | | 63 |

ÍNDICE DE ILUSTRACIONES

FIGURAS

| | | |
|-----|--|----|
| 1. | Tendencia NoSQL..... | 2 |
| 2. | Esquema de búsqueda relacional | 7 |
| 3. | Funcionamiento interno map/reduce | 9 |
| 4. | Escalabilidad horizontal mediante Sharding..... | 13 |
| 5. | Estructura básica archivo YAML | 21 |
| 6. | Paginación en escenario No. 1 | 39 |
| 7. | Acceso a disco físico en escenario No. 1 | 40 |
| 8. | Uso de procesador en escenario No. 1 | 41 |
| 9. | Consumo de memoria RAM en escenario No. 1 | 42 |
| 10. | Paginación en escenario No. 2 | 43 |
| 11. | Acceso a disco físico en escenario No. 2..... | 44 |
| 12. | Uso de procesador en escenario No. 2 | 45 |
| 13. | Consumo de memoria RAM en escenario No. 2 | 46 |
| 14. | Porcentaje de errores de inserción en escenario No. 1..... | 48 |
| 15. | Porcentaje de errores de inserción en escenario No. 2..... | 49 |
| 16. | Porcentaje de errores de acceso a base de datos No. 1..... | 51 |
| 17. | Porcentaje de errores de acceso a base de datos No. 2..... | 52 |
| 18. | Porcentaje de éxito en implementación de solución | 54 |

TABLAS

| | | |
|-----|----------------------------|----|
| I. | Categorías NoSQL..... | 4 |
| II. | Tiempos de inserción | 11 |

| | | |
|--------|---|----|
| III. | Tiempos de recuperación | 11 |
| IV. | Disney Interactive Media Group..... | 22 |
| V. | Foursquare | 23 |
| VI. | MTV networks..... | 24 |
| VII. | Costos de licenciamiento bases de datos NoSQL documentales..... | 26 |
| VIII. | Costos de capacitación de personal de bases de datos NoSQL documentales | 27 |
| IX. | Costos de soporte bases de datos NoSQL documentales (\$) | 28 |
| X. | Propuesta de presupuesto (\$) | 29 |
| XI. | Especificaciones técnicas base de datos PostgreSQL | 32 |
| XII. | Especificaciones técnicas base de datos MongoDB..... | 33 |
| XIII. | Especificaciones técnicas base de datos MySQL..... | 33 |
| XIV. | Especificaciones de escenarios..... | 34 |
| XV. | Estructura de la tabla/colección para los escenarios | 35 |
| XVI. | Variables para evaluación | 35 |
| XVII. | Tiempos de inserción escenario No. 1..... | 37 |
| XVIII. | Tiempos de inserción escenario No. 2..... | 37 |
| XIX. | Consumo de recursos escenario No. 1..... | 38 |
| XX. | Consumo de recursos escenario No. 2..... | 43 |
| XXI. | Errores de inserción en escenario No. 1..... | 47 |
| XXII. | Errores de inserción en escenario No. 2..... | 49 |
| XXIII. | Errores de acceso a base de datos en escenario No. 1 | 50 |
| XXIV. | Errores de acceso a base de datos en escenario No. 2 | 52 |
| XXV. | Resumen de pruebas | 53 |

LISTA DE SÍMBOLOS

| Símbolo | Significado |
|----------------|---------------------------------|
| DB | Base de datos |
| BBDD | Bases de datos |
| \$ | Dólar estadounidense |
| GB | Gigabyte |
| GUI | Interfaz gráfica de usuario |
| MB | Megabyte |
| MHz | Megahercio |
| ms | Milisegundos |
| DBMS | Sistema gestor de base de datos |
| TB | Terabyte |

GLOSARIO

| | |
|-----------------------|---|
| ACID | Conjunto de características necesarias para que una serie de instrucciones en una base de datos sean consideradas como una transacción. ACID es el acrónimo de atomicidad, consistencia, aislamiento y durabilidad. |
| Cluster | Conjunto conglomerado de computadoras construidos mediante la utilización de hardware común y el cual se comporta como si fuesen una única computadora. |
| Código abierto | Open Source es el término con el que se conoce al software que se desarrolla y distribuye de forma libre. |
| CSV | Fichero separado por comas (Comma-Separated Values) utilizado para la representación de datos en forma de tabla, en el cual las filas se separan por saltos de línea y las columnas mediante comas. |
| Float | Tipo de variable, la cual almacena números decimales de hasta 4 bytes. |
| JavaScript | Lenguaje de programación interpretado, implementado en un navegador web, permitiendo así mejoras en la interfaz de usuario y páginas web dinámicas. |

| | |
|------------------|--|
| JSON | Objeto de notación javascript (JavaScript Object Notation), es un formato ligero utilizado para el intercambio de datos basado en javascript el cual no requiere del uso de XML. |
| Join | Es una sentencia de SQL, la cual permite realizar la combinación de registros de dos o más tablas en una base de datos relacional. |
| Paradigma | La forma de pensar de algunas personas, la aceptación de nuevas ideas, pensamientos, creencias incorporadas, que por lo general fueron impuestas durante la niñez, las cuales se aceptan como verídicas o falsas sin necesidad de analizarlas. |
| RAID | Hace referencia a un sistema de almacenamiento, que utiliza múltiples discos duros entre los que se distribuyen o replican los datos. |
| RAM | Memoria de acceso aleatorio, es la memoria en la cual el procesador recibe las instrucciones y guarda los resultados obtenidos, es temporal y volátil. |
| Servidor | Físicamente hablando es una computadora o CPU el cual forma parte de una red de computadoras con la finalidad de proveer de diversos servicios a las demás computadoras conocidas como clientes. |

| | |
|--------------------|--|
| Transacción | Conjunto de órdenes o instrucciones que se ejecutan de forma atómica o indivisible. |
| Tupla | Sinónimo de fila en una tabla. |
| W3C | World Wide Web Consortium, Consorcio internacional que produce recomendaciones para la red de informática mundial. |
| XML | Lenguaje de marcas extensibles (extensible Markup Language) desarrollado por el W3C, para permitir la compatibilidad entre sistemas para compartir información de una manera segura, fácil y fiable. |
| YAML | Lenguaje utilizado para serializar datos basado en XML. |

RESUMEN

El presente trabajo consiste en la investigación de los conceptos necesarios, para realizar el análisis de rendimiento de las bases de datos NoSQL documentales y de tipo relacional, a través de pruebas de estrés que permitan la obtención de datos para análisis e interpretación.

El primer capítulo contiene los conceptos básicos necesarios para entender de mejor manera el surgimiento, manejo de tareas, diferencias con respecto a las bases de datos relacionales, usos y casos de éxito de las bases de datos NoSQL documentales.

El segundo capítulo posee los factores necesarios a considerar para la implementación de una base de datos NoSQL documental, entre los cuales se encuentran: el grado de adaptación, la facilidad de aprendizaje, grado de facilidad de implementación y migración de datos.

En el tercer capítulo se enumeran los costos de implementación de una base de datos documental NoSQL, entre los cuales se encuentran los costos de aplicación como: licencias, capacitación, migración, entre otros, los costos de infraestructura en el caso que se usen varios servidores, RAIDS, etc.

Por último, en el cuarto capítulo se describen las pruebas de estrés correspondientes a la base de datos NoSQL y a las relacionales, para medir y analizar el rendimiento de ambos tipos, tomando en cuenta factores como: tiempo de respuesta, consumo de recursos, escalabilidad, etc.

OBJETIVOS

General

Realizar un análisis comparativo entre el rendimiento proporcionado por una base de datos NoSQL documental y una de tipo relacional, demostrando que las bases de datos NoSQL documentales son la mejor opción en la creación de sitios web de alto rendimiento.

Específicos

1. Analizar el impacto de las bases de datos NoSQL documentales en el rendimiento de los sitios web que las implementan, tomando como base MongoDB 10gen.
2. Proporcionar un punto de referencia para el análisis de la facilidad, para implementar una base de datos NoSQL documental en un sitio web, basándose en costos de implementación y en el rendimiento del sitio web.
3. Demostrar que una base de datos NoSQL documental es mejor para el alto rendimiento de un sitio web, mediante el análisis y comparación de costos y de rendimiento entre una base de datos relacional y una NoSQL documental.

INTRODUCCIÓN

En la actualidad, el objetivo de toda empresa o negocio es el de proporcionar servicios de alta calidad a un bajo costo, esto con la finalidad de obtener prestigio, fama, remuneración económica, entre otros, por lo cual un servicio que no posea como mínimo: disponibilidad, respuesta a fallos, rapidez en las transacciones, por mencionar algunas; está destinado a fracasar, en otras palabras se necesita de sitios web que posean alto rendimiento.

Las bases de datos NoSQL manejan el almacenamiento de la información de una forma llave-valor, esto quiere decir, que las tablas son dinámicas, lo cual implica que al momento de ocurrencia de un cambio, este no afecta la información, es decir, se acepta tener un registro poseedor de más campos o menos campos que los similares, esto permite cambiar de forma rápida la estructura de la información sin necesidad de rediseñar todo el modelo de nuevo.

Dentro de las bases de datos NoSQL se encuentran las del tipo documental, también llamadas orientadas a documentos, estas se encuentran diseñadas y estructuradas para el almacenamiento, recuperación y gestión de información que no posee una estructura definida, la cual se encuentra almacenada en documentos tipo JSON, debido a esto no es necesario el uso de claves u otros identificadores únicos definidos por el programador, por la simple razón que la misma base de datos les asigna un valor único e inequívoco para control.

1. BASES DE DATOS NOSQL DOCUMENTALES

Este capítulo contiene información acerca del surgimiento de las bases de datos NoSQL y las NoSQL documentales, de igual manera los conceptos relacionados a las bases de datos NoSQL, para entender el funcionamiento y la aplicación en ambientes de producción.

1.1. Surgimiento de las bases de datos NoSQL

A principios de 1970, como parte de un experimento en las universidades europeas y norteamericanas, se desarrollaron sistemas de almacenamiento abierto.

En 1998, Carlo Strozzi, fue el primero en utilizar el término NoSQL refiriéndose a bases de datos que eran desarrolladas bajo código abierto, ligeras y que no ofrecían interfaces como las bases de datos convencionales, sin embargo, usaban un modelo relacional.

Más tarde en 1999, Eric Evans, programador de Rackspace, adoptó el término NoSQL en un evento sobre bases de datos distribuidas de código abierto.

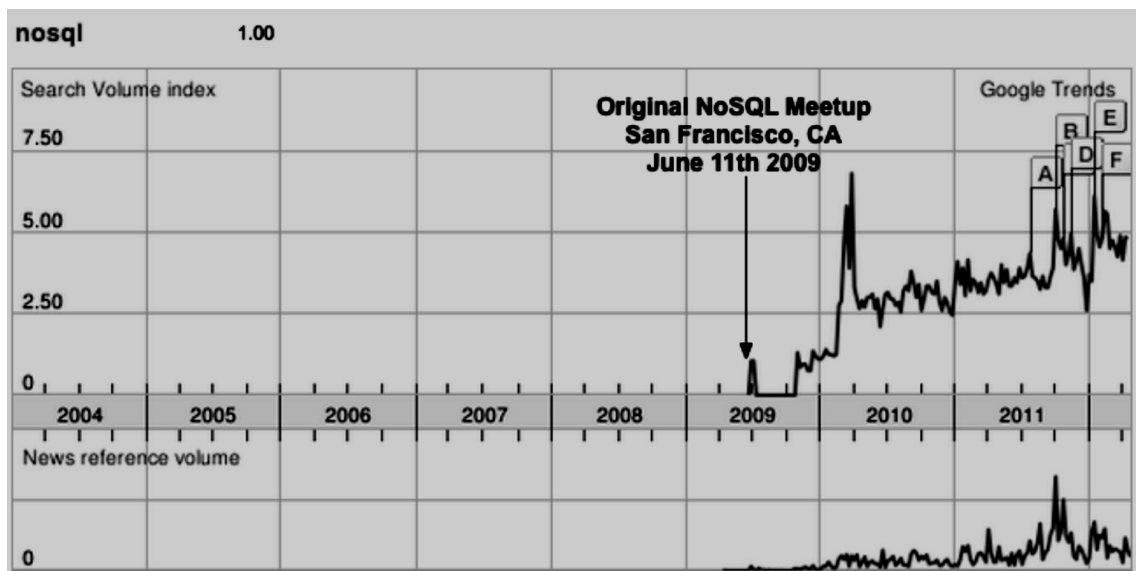
En el 2004 surgió la primera implementación de un sistema de bases de datos NoSQL, cuando Google lanzó la famosa BigTable, la cual posee un alto rendimiento, esto le permitió proveer un sinnúmero de aplicaciones, tales como: Maps o Google Earth.

Durante el 2007, con la aparición de Dynamo por parte de Amazon, se marca un momento importante dentro de la historia de las bases de datos NoSQL, la cual es de código propietario.

A principios del 2008 se inicia otro proyecto importante en bases de datos NoSQL, Cassandra, el cual es un sistema de bases de datos NoSQL, que se caracteriza por la gran disponibilidad, éste fue implementado por Facebook y luego sustituyó a MySQL como base de datos en Twitter.

En la actualidad se ha difundido el uso de bases de datos NoSQL, como se muestra en la figura 1, debido a la utilidad para la creación de redes sociales, sitios de comercio electrónico, etc., como por ejemplo: Amazon, Twitter, Facebook y Google.

Figura 1. **Tendencia NoSQL**



Fuente: <http://www.google.com/trends>. Consulta: noviembre de 2012.

1.2. Bases de datos NoSQL

Las bases de datos NoSQL son aquellas que no poseen un esquema entidad relación y no utilizan SQL como el lenguaje principal para la realización de consultas. Entre las características más importantes de las bases de datos NoSQL se encuentran:

- No-relacionales
- Distribuidas
- Código abierto
- Escalabilidad horizontal

Las bases de datos NoSQL no poseen esquemas de usuarios, estructuras como tablas, no permiten la realización de uniones en las consultas y no garantizan ACID, cuentan con una cantidad considerable de categorías, las cuales poseen características específicas que proporcionan soluciones óptimas y eficientes ante diversos problemas.

A continuación se muestra la tabla I, en la que se resumen las características de las distintas categorías de bases de datos NoSQL existentes en la actualidad y las principales bases de datos que se encuentran en cada categoría.

Tabla I. **Categorías NoSQL**

| Categoría | Características | Ejemplos |
|---------------------|--|---|
| Familia de columnas | <ul style="list-style-type: none"> • Clave por cada columna. • Una tupla es equivalente a una columna. | <ul style="list-style-type: none"> • Hadoop • Cassandra • Stratosphere |
| Clave-valor | <ul style="list-style-type: none"> • Datos almacenados en hashmap. • Cada elemento posee llave única. | <ul style="list-style-type: none"> • DynamoDB • Redis • Scalaris |
| Documentales | <ul style="list-style-type: none"> • Información almacenada en documentos. • Escalabilidad y alto rendimiento. | <ul style="list-style-type: none"> • MongoDB • CouchDB • RavenDB |
| Orientadas a grafos | <ul style="list-style-type: none"> • La información se almacena en nodos. • Las relaciones son aristas y vértices. | <ul style="list-style-type: none"> • Neo4J • InfoGrid • DEX |
| Orientada a objetos | <ul style="list-style-type: none"> • Integra lenguajes de programación y no solamente SQL. | <ul style="list-style-type: none"> • Db4o • Versant • Perst |
| Bases XML | <ul style="list-style-type: none"> • Los datos se almacenan en formato XML y se complementan con las documentales. | <ul style="list-style-type: none"> • eXist • Sedna • Qizx |

Fuente: <http://nosql-database.org/>. Consulta: noviembre de 2012.

1.3. Bases de datos NoSQL documentales

Una base de datos NoSQL documental almacena la información en forma de documento, utilizando punteros para el acceso de la información almacenada en la base de datos, esto se realiza sin la ayuda de una estructura entidad relación predeterminada, como en el caso de las bases de datos relacionales. La búsqueda de la información se realiza con base al contenido del documento.

1.4. Comparativa DBMS relacional versus DBMS NoSQL

Para realizar una comparativa se deben analizar los aspectos de funcionamiento de un DBMS según la naturaleza de la bases de datos y los esquemas de búsqueda de información de los mismos, en los siguientes apartados se detallan los aspectos mencionados anteriormente.

1.4.1. Función general de un DBMS

La función general de un DBMS es manejar de forma segura, sencilla y ordenada los datos que posteriormente se convertirán en información relevante para una organización. También son utilizados como interfaz para la comunicación entre la base de datos, el usuario y las aplicaciones que se utilizan. Un DBMS debe cumplir con ciertos aspectos los cuales son:

- Abstracción de la información
- Independencia
- Consistencia
- Seguridad
- Transacciones
- Tiempo aceptable de respuesta

En los siguientes apartados se describen, de forma general, los DBMS del tipo relacional y los DBMS de tipo NoSQL, de igual manera se analizan las diferencias entre ambos.

1.4.1.1. DBMS relacional

Un DBMS relacional ofrece a los usuarios validación de procesos, garantía y verificación de integridad de datos, control de concurrencia, recuperación de fallas, seguridad, control de transacciones, optimización de consultas, entre otros. La utilización de este tipo de DBMS facilita, en gran manera, el manejo de la base de datos al usuario.

1.4.1.2. DBMS NoSQL

En el caso de los DBMS NoSQL, al estar diseñados para soportar una estructura más flexible, brindan mayor capacidad de respuesta ante un gran volumen de solicitudes para escritura o lectura de datos sin estar sujetos a las restricciones de un modelo relacional, como es el uso de llaves primarias.

1.4.1.3. Diferencia entre DBMS relacional y DBMS NoSQL

Debido a la naturaleza de las bases de datos relacionales y NoSQL (llamadas no relacionales), se observan las siguientes diferencias a nivel de los DBMS:

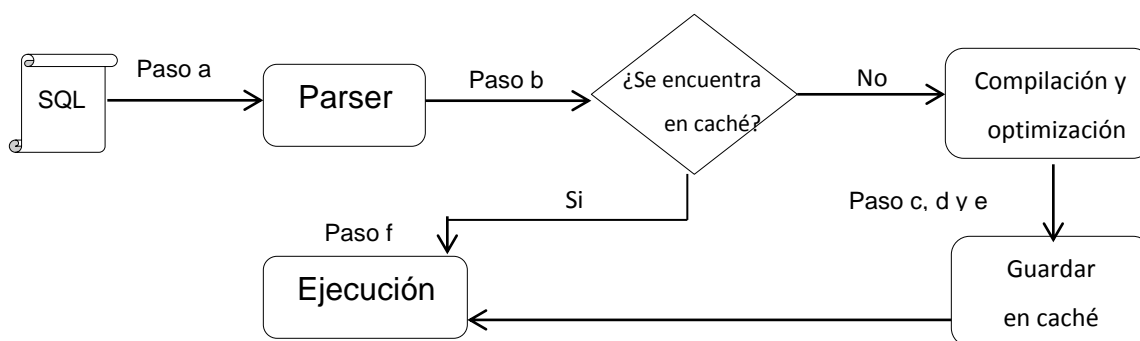
- Los DBMS relacionales poseen verificación de integridad de datos (características ACID), mientras que los DBMS NoSQL no poseen dicha verificación.
- Los DBMS NoSQL mejoran el rendimiento de la lectura y escritura de datos.

- La flexibilidad en las estructuras de la base de datos es obtenida únicamente al momento de utilizar un DBMS NoSQL.
- El consumo de recursos del equipo es considerablemente bajo al utilizar los DBMS NoSQL.¹

1.4.2. Esquema de búsqueda de datos en DBMS relacional

Un esquema de búsqueda de datos es la forma de analizar y organizar la información obtenida de una o varias estructuras de una base de datos, en la figura 2 se encuentran representados los pasos que un DBMS realiza al momento de interpretar una consulta.

Figura 2. Esquema de búsqueda relacional



Fuente: elaboración propia.

¹ Universidad de Fortaleza [http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840 Bancos%20de%20Datos%20NoSQL.pdf](http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840Bancos%20de%20Datos%20NoSQL.pdf). Consulta: noviembre de 2012.

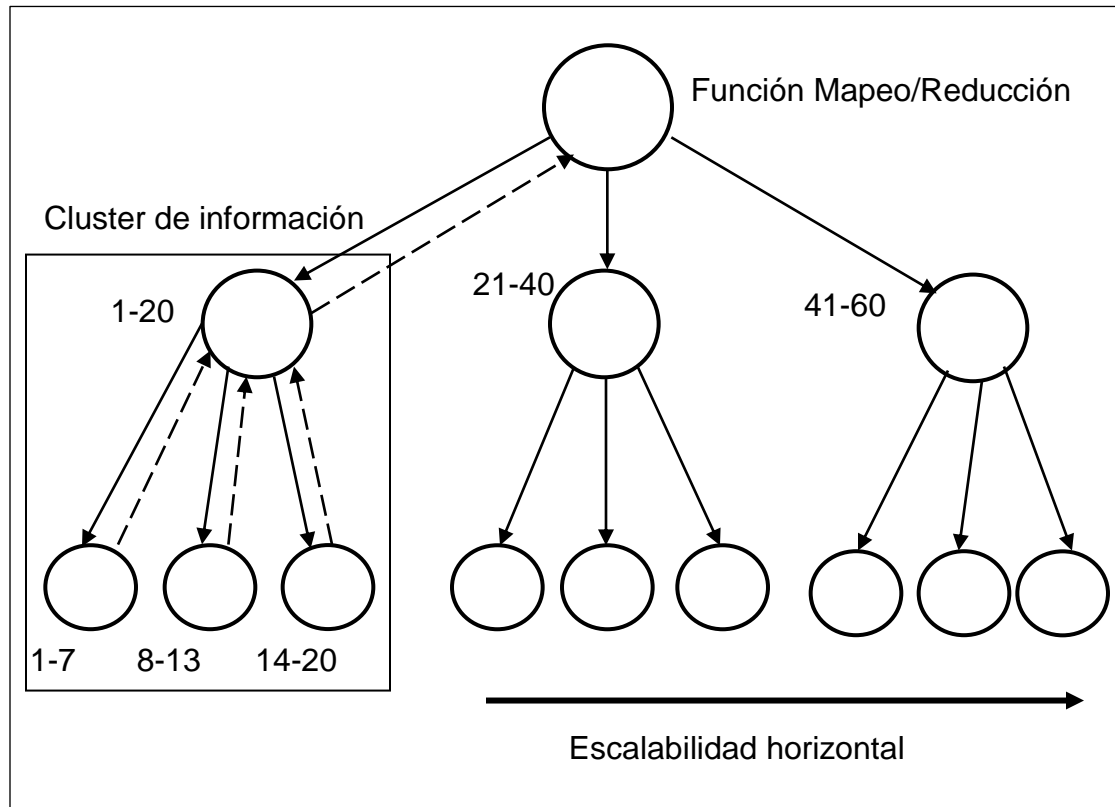
Como se observa en la figura 2, el proceso de búsqueda de información realizado por un DBMS relacional consta de una serie de pasos, los cuales se describen a continuación:

- Lectura de la consulta: consiste en evaluar la sentencia escrita en lenguaje SQL, verificando que no posea errores de redacción.
- Una vez definidos los datos a obtener se procede a definir cómo se obtendrá la información, se realiza una lista que contiene de forma ordenada y detallada los pasos a seguir para la obtención de los datos: tablas a consultar, índices empleados, la forma de unir la información de las distintas tablas, etc.
- En el caso que la consulta ya se haya realizado anteriormente y se encuentre en caché se procede a ejecutarla.
- Se realiza la compilación de la consulta varias veces, lo que implica la generación de varios planes de ejecución.
- Cuando se ha elegido el plan de ejecución más eficiente, se procede al almacenamiento en una caché de planes de ejecución.
- Se ejecuta la sentencia SQL.

1.4.2.1. Esquema de búsqueda NoSQL Documental

En el caso de las bases de datos NoSQL documentales, el proceso de búsqueda se realiza con base en un proceso de map/reduce, el cual se observa en la figura 3.

Figura 3. **Funcionamiento interno map/reduce**



Fuente: MÉNDEZ, Sergio. Conferencia sobre NoSQL USAC 2012.

Como se observa en la figura 3, el proceso de mapeo representado por la línea continua, consiste en ubicar la información en el nodo o nodos y el proceso de reducción, representado por la línea punteada, consiste en devolver la información deseada.

A continuación se describe el proceso de map/reduce:

- El usuario solicita un dato a la base de datos NoSQL.

- El controlador de la base de datos NoSQL consulta el rango de claves a los nodos hijos, para ubicar el nodo que posee la información solicitada.
- Se realiza una búsqueda en cada uno de los hijos hasta encontrar la información solicitada.
- Al momento de encontrar la información deseada se procede a realizar el reduce, el cual consiste en devolver la información encontrada a cada uno de los nodos padres hasta llegar al nodo principal y mostrar la información al usuario.

1.5. Evaluación de requerimientos no funcionales para bases de datos relacionales versus bases de datos NoSQL documentales

En el presente apartado se muestran evaluaciones basadas en estudios previos realizados por empresas y organizaciones como: Amazon y Ubuntu.

1.5.1. Tiempos de inserción y recuperación

Los principales procesos que se manejan en una base de datos NoSQL documental son los de recuperación de información y de inserción de datos.

En la tabla II se observan los resultados de pruebas de inserción realizadas a una base de datos relacional, como el caso de PostgreSQL y una base de datos NoSQL documental como MongoDB 10 gen.

Tabla II. **Tiempos de inserción**

| Tipo de base de datos | Base de datos | Registros insertados | Tiempo total de inserción (en seg.) |
|------------------------------|----------------------|-----------------------------|--|
| Relacional | PostgreSQL 9,08 | 250 000 | 436,18 |
| NoSQL | MongoDB 10gen | 250 000 | 13,82 |

Fuente: Ubuntu-es. <http://www.ubuntu-es.org/node/153667>. Consulta: noviembre de 2012.

La prueba para esta operación consistió en la inserción de 250 000 registros en una tabla que poseía 3 campos, 2 de tipo cadena y uno de tipo numérico, las bases de datos en las que se ejecutó la prueba fueron MongoDB como base de datos NoSQL documental y PostgreSQL como base de datos relacional, ambas pruebas se realizaron en una computadora con sistema operativo Ubuntu en la versión 10,04 con 2 GB de RAM.

La tabla III muestra los resultados obtenidos al momento de realizar la recuperación de registros de una base de datos relacional representada por PostgreSQL y una base de datos NoSQL representada por Dynamo.

Tabla III. **Tiempos de recuperación**

| Tipo de base de datos | Base de datos | Registros recuperados | Tiempo total de recuperación de registros (en ms.) |
|------------------------------|----------------------|------------------------------|---|
| Relacional | PostgreSQL 9,08 | 118 | 4000 |
| NoSQL | Dynamo | 118 | 300 |

Fuente: Amazon Dynamo. <http://www.nosql.es/blog/nosql/amazon-dynamo.html>. Consulta: noviembre de 2012.

Esta prueba se realizó para la operación anterior en la recuperación de 118 registros en una tabla que poseía tres campos; dos de tipo fecha y uno de tipo numérico, las bases de datos en las que se ejecutó la prueba fueron Dynamo como base de datos NoSQL documental y PostgreSQL versión 9,08 como base de datos relacional.

1.5.2. Manipulación de información

Las bases de datos NoSQL documentales almacenan y manipulan la información registrada en ellas, mediante documentos que se guardan en el disco duro del servidor o computadora.

Refiriéndose específicamente, a MongoDB, se encapsula y codifica cada documento almacenado con un formato XML, YAML, JSON o BSON, pero también ofrecen la posibilidad de hacerlo en archivos de tipo PDF o documentos de Microsoft Office como: Word, Excel, por mencionar algunos. Cada documento se almacena en estructuras llamadas colecciones, las cuales son análogas a las tablas de las bases de datos relacionales.

1.5.3. Fiabilidad de transacciones

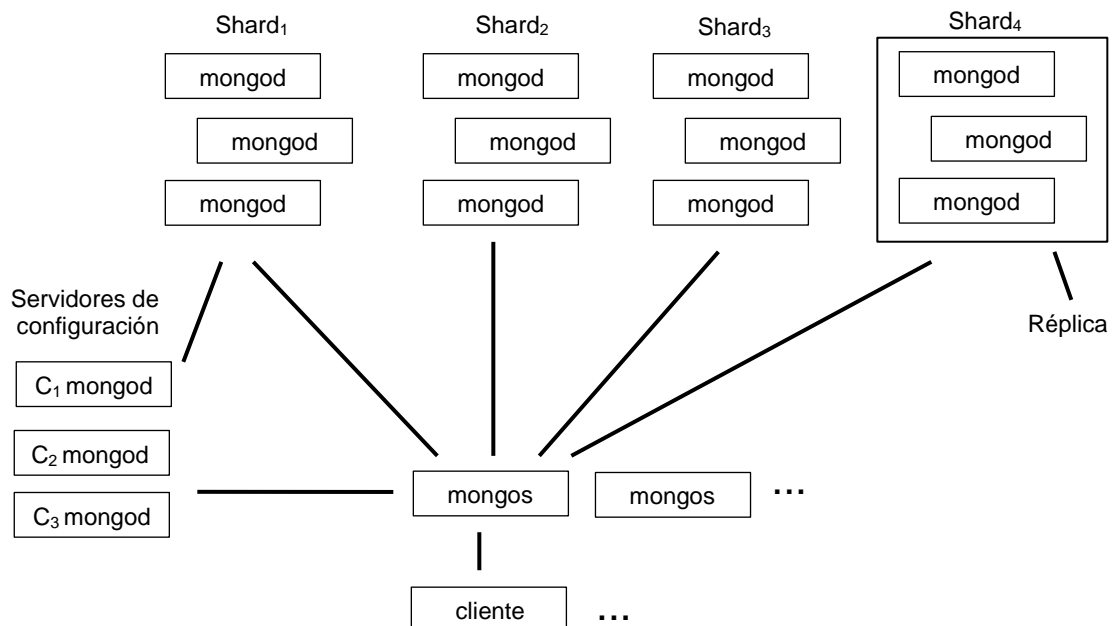
Las bases de datos NoSQL documentales ofrecen gran rendimiento y fiabilidad en cuanto a transacciones se refiere, tienen la capacidad de dar servicio a grandes volúmenes de información de lectura/escritura e indexación de una gran cantidad de documentos.

En el aspecto de la fiabilidad de transacciones, las bases de datos NoSQL documentales aun poseen un grado de inmadurez elevado, con respecto a las bases de datos relacionales. Esto debido a que no poseen ACID, ya que no garantizan la atomicidad, consistencia, aislamiento y durabilidad de los datos.

1.5.4. Escalabilidad y elasticidad a través de técnicas de particionamiento

La escalabilidad es la propiedad que posee un sistema de aumentar la capacidad de trabajo o tamaño sin que se vea comprometido el rendimiento, MongoDB ofrece la opción de Sharding, la cual consiste en distribuir la base de datos en un conjunto de clúster de máquinas, como se observa en la figura 4.

Figura 4. Escalabilidad horizontal mediante Sharding



Fuente: MongoDB sharding. <http://www.kchodorow.com/blog/2010/03/30/sharding-with-the-fishes/>. Consulta: noviembre de 2012.

1.5.5. Alta disponibilidad

Como se ilustra en la figura 4, al realizar Sharding de la base de datos se garantiza que la información esté disponible en cualquier momento, esto debido a que la información se encuentra dividida de forma lógica en distintas máquinas lo que ayuda a mejorar el consumo de recursos disponibles. Lo cual permite que varios usuarios tengan acceso concurrente a los datos, sin sufrir retrasos.

1.6. Ventajas en ambientes de producción utilizando bases de datos NoSQL documentales

Las razones más importantes, por las cuales algunas compañías justifican el uso de las bases de datos NoSQL documentales en lugar de las relacionales, para el almacenamiento de la información son las siguientes:

- Alto rendimiento en escrituras masivas a la base de datos: las bases de datos NoSQL documentales permiten almacenar grandes cantidades de información la cual varía desde 1 TB hasta 70 TB.
- Almacenamiento en caché: la información con más accesos se almacena en caché, la de medio acceso en memoria y la que tienen una tasa baja de acceso en disco.
- No existen puntos de fallo, debido a que se distribuye el flujo de información mediante la implementación de balanceo de carga automático.

- Fácil uso de programación: los datos son accedidos fácilmente, el uso es más intuitivo para los programadores.

1.7. MongoDB como base de datos NoSQL documental

MongoDB presenta una serie de características, las cuales le han ayudado a posicionarse como una base de datos preferencial por las compañías, debido a dichas características MongoDB se adecúa perfectamente para los usos de:

- Juegos en línea
- Comercio electrónico
- Solución de problemas con grandes volúmenes de información
- Sistemas de manejo de documentos y contenidos
- Almacenamiento de datos operacionales de un sitio web
- Almacenamiento de votaciones
- Registros y perfiles de usuarios
- Estadísticas en tiempo real
- Manejo de bitácoras

Es importante resaltar que las bases de datos NoSQL al no manejar ACID no son recomendables para manejar transacciones monetarias importantes o cuentas bancarias, sin embargo, en el caso de MongoDB, hay empresas que la han utilizado en la implementación de transacciones de esa índole.

2. FACTORES A CONSIDERAR PARA IMPLEMENTAR UNA BASE DE DATOS NOSQL DOCUMENTAL

Para implementar una base de datos NoSQL documental en lugar de una relacional, no es suficiente con conocer las ventajas o características que ofrecen, también es necesario considerar el grado de facilidad y grado de adaptación para la implementación.

2.1. Grado de adaptación

La adaptación es una modificación a largo plazo, ya sea en el funcionamiento, procesos o comportamiento de un sistema, la cual se realiza para el mejoramiento del mismo, existen dos factores importantes que influyen en la adaptación, los cuales son el tiempo y la facilidad para aceptar el cambio.

2.1.1. Tiempo medio de aceptación y adaptación

La adaptación y aceptación de las bases de datos NoSQL es muy variada según la empresa o compañía que desee implementarlas, es por eso que el tiempo que el personal se tarda en adaptarse a la nueva tecnología varía de días a semanas o incluso meses.

Existe una gran variedad de factores que influyen de gran manera en la aceptación de una base de datos NoSQL documental, entre los cuales se encuentran:

- Adopción de un nuevo paradigma: esta es una de las causas por las cuales resulta relativamente difícil la adaptación al concepto de NoSQL, ya que la mayoría de compañías se encuentran más familiarizadas con el concepto de bases de datos relacionales.
- Curva de aprendizaje: al ser una tecnología relativamente nueva, esto representa para algunas personas un obstáculo en aprender la utilización de las bases de datos NoSQL.
- Costo-beneficio: es uno de los factores más importantes, debido al hecho de comparar si los beneficios que se obtendrán son mayores a los costos de implementarlo.

Una vez se han superado estos tres factores, la adaptación de las bases de datos NoSQL documentales dependen del conocimiento de javascript que posean los programadores de la empresa. Según estudios realizados por la empresa MongoDB sobre la base de datos MongoDB 10gen, si los programadores poseen conocimientos amplios de javascript la adaptación y aceptación será de 5 a 8 días, de lo contrario la adaptación y aceptación podrá ir de 3 semanas a 6 meses, variando de acuerdo al nivel de conocimiento del lenguaje.²

En el siguiente apartado se hace referencia al nivel de la curva de aprendizaje que poseen las bases de datos NoSQL documentales, el cual es un factor decisivo en el tiempo de aceptación de dicha tecnología.

² <http://www.10gen.com/products/training>. Consulta: noviembre de 2012.

2.1.2. Nivel de curva de aprendizaje

La curva de aprendizaje describe de forma gráfica el grado o cantidad de conocimiento adquirido con éxito durante un periodo de tiempo de aprendizaje determinado, en el eje horizontal se representa el tiempo que ha transcurrido y en el eje vertical se representa la cantidad de conocimiento adquirido.

El nivel de la curva de aprendizaje de las bases de datos NoSQL documentales se encuentra estrechamente relacionado con el nivel de manejo de java script que se posea, esto a causa que las bases de datos de este tipo almacenan toda la información en archivos JSON, los cuales poseen una estructura similar a JavaScript pero sin XML.

2.2. Grado de facilidad de implementación

La facilidad al momento de implementar una solución basada en una base de datos NoSQL se ve afectada por varios factores, entre los cuales se mencionan los siguientes: requisitos mínimos para implementar la solución y la portabilidad con respecto a la migración de datos, los cuales se amplían a continuación:

2.2.1. Requisitos mínimos de implementación

De acuerdo a los requerimientos de MongoDB se debe poseer como mínimo, los siguientes requerimientos para la implementación de una base de datos NoSQL documental.³

³ <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>. Consulta: noviembre de 2012.

- Hardware
 - Computadoras con procesadores Pentium 4 o mejores.
 - Caché de nivel 2 (L2) de 2 MB.
 - Memoria RAM DDR-2 de 2 GB de capacidad o superior.
 - Disco duro de 120 GB de capacidad o superior.
 - Tarjeta madre con velocidad de bus de 800 MHz y memoria de 667 MHz.
 - Todo el hardware debe ser de 64 bits.

- Software
 - El DBMS que se desee utilizar.
 - Un sistema operativo de 64 bits, no importando si es Oses, Linux, Mac, Windows, Solaris, etc.

2.3. Portabilidad respecto a la migración de datos

La forma más común o utilizada en la actualidad, para realizar migraciones de bases de datos del tipo relacional hacia una NoSQL documental es mediante la utilización de archivos YAML. Es importante notar que no es la única forma de realizar la migración de información, pero es el más utilizado por las compañías.

En los siguientes incisos se describe el proceso general para realizar la migración de los datos mediante el uso de archivos YAML.

- Exportar datos a un archivo CSV, al momento de importar la información a MongoDB se tomará como una sola cadena sin hacer distinción de enteros, fechas, punto flotante, etc.
- Realizar la migración de información a través de archivos YAML.
- Ejecutar runDumper.php, el cual es el archivo YAML con las especificaciones de los datos.
- Enumerar tablas o campos para la omisión de la carga dentro del archivo de configuración.
- Mediante el uso de PHPMigrationMySQL se asignan los tipos básicos de datos a MongoDB.
- Se realiza la importación del archivo YAML a MongoDB.

Después de haber descrito los pasos generales de la migración es importante describir la estructura básica de un archivo YAML, para la migración de datos, dicha estructura es la siguiente:

Figura 5. **Estructura básica archivo YAML**

```
tablas: usuario:
  many_references:
    reference_table: user_group
    in_id: user_id
    out_id: group_id
    out_table: grupo
```


Fuente: <http://www.yaml.org/spec/1.2/spec.html>. Consulta: noviembre de 2012.

2.4. Casos de éxito de bases de datos NoSQL documentales

Es importante mencionar los casos de éxito en la utilización de bases de datos NoSQL documentales por parte de las compañías, debido al prestigio de dichas compañías y los beneficios obtenidos gracias a la implementación de soluciones NoSQL documentales.

Las compañías que han implementado bases de datos NoSQL documentales han obtenido beneficios sustanciales en el rendimiento de los sitios web, a continuación se detalla el problema y la respectiva solución mediante la implementación de una base de datos NoSQL documental:


Tabla IV. **Disney Interactive Media Group**

| Aspecto | Descripción |
|---|---|
|  | Es una de las compañías de entretenimiento con mayor auge en la actualidad ofreciendo juegos interactivos, los cuales son utilizados por millones de personas cada día. |
| El problema | Cada aplicación y servicio en el sitio web poseía un modelo de datos personalizado. |
| Solución | Centralizar los datos en una sola plataforma con mayor escalabilidad y flexibilidad mediante la implementación de MongoDB 10gen. |

Fuente: MongoDB 10gen. <http://www.10gen.com/customers/disney>. Consulta: noviembre de 2012.

Con la implementación de MongoDB 10gen, por parte de Disney, se obtuvo la independencia en el almacenamiento de la información y el desarrollo de las aplicaciones lo cual proporciona un mayor control sobre los datos.


Tabla V. **Foursquare**

| Aspecto | Descripción |
|---|---|
|  | <p>Red social basada en la ubicación geoposicional o GPS de los suscriptores, con lo cual se permite al usuario realizar publicaciones desde móviles de todos los lugares que han visitado.</p> |
| <p>El problema</p> | <p>Se contaba con una única base de datos relacional, en la cual se almacenaba toda la información de los usuarios, provocando problemas de escalabilidad.</p> |
| <p>Solución</p> | <p>Migrar toda la información a una base de datos NoSQL documental.</p> |

Fuente: MongoDB 10gen. <http://www.10gen.com/customers/foursquare>. Consulta: noviembre de 2012.

La solución implementada por Foursquare provocó, no solamente la solución de la problemática vigente hasta ese momento, también proporcionó mayor capacidad de replicación, una mayor capacidad de accesos concurrentes por parte de los usuarios y una mayor flexibilidad de almacenamiento, todo esto en conjunto ayudó a mejorar el rendimiento del sitio web.

Tabla VI. **MTV networks**

| Aspecto | Descripción |
|---|--|
|  | <p>Compañía propietaria de una gran cantidad de sitios web que manejan cantidades elevadas de usuarios, entre las cuales se encuentran: skype.com, nick.com y gametrailers.com.</p> |
| <p>El problema</p> | <p>Todos los sitios web propiedad de la compañía MTV networks fueron creados con un fin comercial y no pensando en soportar una gran cantidad de usuarios como en la actualidad, lo que provocó el colapso de las bases de datos.</p> |
| <p>Solución</p> | <p>Se realizó una migración masiva de la información a una base de datos NoSQL documental, para obtener mayor flexibilidad y una mejor escalabilidad horizontal.</p> |

Fuente: MongoDB 10gen. <http://www.10gen.com/customers/mtv-networks>. Consulta: noviembre de 2012.

Al momento de la implementación por parte de MTV Networks de la solución ofrecida por MongoDB 10gen, se observó un aumento notable en la escalabilidad horizontal y un mejor manejo de grandes volúmenes de información concurrente.

3. COSTOS DE IMPLEMENTACIÓN

Dentro de las bases de datos NoSQL documentales existen dos grupos los cuales son: las gratuitas y las que necesitan el pago de licenciamiento para funcionar. En el presente capítulo se especifican los costos de implementación necesarios para distintas bases de datos NoSQL documentales y los costos de infraestructura necesarios para el correcto funcionamiento.

3.1. Costos de aplicación

Las bases de datos NoSQL documentales poseen, entre una de las muchas ventajas, el costo monetario, la mayoría de soluciones NoSQL documentales poseen licenciamiento de código no propietario, lo que permite descargarlas y hacer uso de las funciones sin ningún costo.

3.1.1. Licenciamiento

El licenciamiento es un tratado entre dos o más entidades conocidos como licenciante o usuario y el licenciataro o dueño, en el cual se encuentran las normas o especificaciones que rigen el uso, distribución, instalación o copias de la aplicación de software.⁴

Existen dos categorías de licenciamiento de software:

⁴ <http://www.licencias.cl/index.php?smenu=Licencias%20OEM-OLP>. Consulta: noviembre de 2012.

- Código no propietario: permite la ejecución del programa para cualquier propósito que se desee, redistribución de copias o poner a disposición del público mejoras realizadas para beneficio de toda la comunidad.
- Código propietario: los términos de uso, distribución, copia, modificación o cualquier redistribución total o parcial de la aplicación es establecida por el propietario de los derechos de autor de la aplicación.

En la tabla VII se listan los los costos de licenciamiento de las bases de datos NoSQL documentales de mayor aceptación por las compañías actualmente.

Tabla VII. **Costos de licenciamiento bases de datos NoSQL documentales**

| DBMS | Licenciamiento | Costo (USD) | Período de licenciamiento |
|--|-----------------------|--------------------|----------------------------------|
| DynamoDB | Propietario | 0,4 TB/mes | Mensual |
| CouchDB | No propietario | 0,00 | Tiempo indefinido |
| Hadoop | No propietario | 0,00 | Tiempo indefinido |
| MongoDB 10gen | No propietario | 0,00 | Tiempo indefinido |
| Oracle NoSQL Database Enterprise Edition | Propietario | 5 897,38 | Tiempo indefinido |

Fuente: DynamoDB, CouchDB, Hadoop, MongoDB, Oracle. Consulta: noviembre de 2012.

3.1.2. Capacitación

Al momento de realizar la implementación de nuevas tecnologías es importante realizar la capacitación del personal, para incrementar los beneficios que ofrecen las nuevas soluciones y no por el contrario, que representen un retraso en el desempeño del empleado.

En la tabla VIII se observa un resumen de los costos de capacitación en diversas bases de datos NoSQL documentales y de igual manera la duración de las mismas.

Tabla VIII. **Costos de capacitación de personal de bases de datos NoSQL documentales**

| DBMS | Duración | Costo por persona (\$) |
|---------------|---------------|------------------------|
| DynamoDB | NA | NA |
| CouchDB | 2 días | 1 215,58 |
| Hadoop | 4 días | 3 437,05 |
| MongoDB 10gen | 2 días | 1 519,47 |
| Oracle NoSQL | No disponible | No disponible |

Fuente: DynamoDB⁵, CouchDB⁶, Hadoop⁷, MongoDB⁸, Oracle⁹. Consulta: noviembre de 2012.

3.1.3. Soporte

El soporte se encuentra dividido en dos grupos principales de acuerdo al tipo de asistencia proporcionada al cliente, los cuales son:

⁵ <http://aws.amazon.com/es/dynamodb/pricing/>. Consulta: noviembre de 2012.

⁶ <http://couchdb.apache.org/>. Consulta: noviembre de 2012.

⁷ <https://ccp.cloudera.com/display/SUPPORT/Downloads>. Consulta: noviembre de 2012.

⁸ <http://www.mongodb.org/downloads>. Consulta: noviembre de 2012.

⁹ <https://shop.oracle.com/pls/ostore/f?p=700:6:0::NO:::>. Consulta: noviembre de 2012.

- Soporte comunidad: es el proporcionado sin ningún costo adicional y en cualquier momento. Los medios más utilizados para la realización de este tipo de soporte son: foros, blogs, correos, canales de internet.
- Soporte empresarial: es el proporcionado por las empresas el cual incurre en un costo, que por lo general, no es incluido en costo del software, los medios más utilizados son: llamadas telefónicas, video conferencias, asistencia presencial y asistencia en línea.

A continuación se presenta un resumen de los costos de soporte técnico en los cuales se incurre al momento de adquirir una base de datos NoSQL.

Tabla IX. **Costos de soporte bases de datos NoSQL documentales (\$)**

| DBMS | Tipo de soporte técnico | Horario de atención | Costo por mes |
|---------------|--|---------------------|---------------|
| DynamoDB | Correo electrónico | 24 horas, 365 días | 40,19 |
| CouchDB | Foros y correo electrónico | NA | 0,00 |
| Hadoop | Correo electrónico | 24 horas, 365 días | 0,00 |
| MongoDB 10gen | Teléfono y correo electrónico | 24 horas, 365 días | 33,49 |
| Oracle NoSQL | Teléfono, correo electrónico y live chat | 24 horas, 365 días | 184,55 |

Fuente: DynamoDB¹⁰, CouchDB¹¹, Hadoop¹², MongoDB¹³, Oracle¹⁴. Consulta: noviembre de 2012.

¹⁰ <http://aws.amazon.com/es/dynamodb/pricing/>. Consulta: noviembre de 2012.

¹¹ <http://couchdb.apache.org/>. Consulta: noviembre de 2012.

¹² <https://ccp.cloudera.com/display/SUPPORT/Downloads>. Consulta: noviembre de 2012.

¹³ <http://www.mongodb.org/downloads>. Consulta: noviembre de 2012.

¹⁴ <https://shop.oracle.com/pls/ostore/f?p=700:6:0::NO:::>. Consulta: noviembre de 2012.

Es importante resaltar, que únicamente se tomaron en cuenta las bases de datos que poseen mayor aceptación y utilización en la actualidad.

3.2. Costos de infraestructura

Los costos de infraestructura son dependientes de aspectos, tales como: el uso asignado a la base de datos, tamaño de la compañía u objetivo de la implementación.

La tabla X propone el presupuesto concerniente a los gastos de una empresa mediana la cual no necesita de una infraestructura compleja o demasiado costosa para el funcionamiento.

Tabla X. Propuesta de presupuesto (\$)

| Aspecto | DynamoDB | CouchDB | Hadoop | MongoDB 10gen | Oracle NoSQL |
|--------------------|----------|---------|-----------|------------------|-----------------|
| Licenciamiento | 502,40 | 0 | 0 | 0 | 4 807,70 |
| Agregados | 0 | 0 | 0 | 0 | 0 |
| Capacitación | 0 | 1215,58 | 27 365,04 | 1519,47 | 0 |
| Soporte técnico | 40,19 | 0 | 0 | 0 | 184,55 |
| Total | 542,59 | 1215,58 | 27 365,04 | 1519,47 | 4 992,22 |

Fuente: DynamoDB¹⁵, CouchDB¹⁶, Hadoop¹⁷, MongoDB¹⁸, Oracle¹⁹. Consulta: noviembre de 2012.

¹⁵ <http://aws.amazon.com/es/dynamodb/pricing/>. Consulta: noviembre de 2012.

¹⁶ <http://couchdb.apache.org/>. Consulta: noviembre de 2012.

¹⁷ <https://ccp.cloudera.com/display/SUPPORT/Downloads>. Consulta: noviembre de 2012.

¹⁸ <http://www.mongodb.org/downloads>. Consulta: noviembre de 2012.

¹⁹ <https://shop.oracle.com/pls/ostore/f?p=700:6:0::NO::>. Consulta: noviembre de 2012.

4. PRUEBAS DE RENDIMIENTO BASE DE DATOS NOSQL DOCUMENTAL VERSUS SQL RELACIONAL

En los capítulos anteriores se mencionaron las características y ventajas que las bases de datos NoSQL documentales poseen, las cuales las convierten en una buena opción para la implementación de sitios web comparados con las bases de datos relacionales, pero ¿es correcto afirmar esto?, este capítulo contiene el detalle de las pruebas realizadas para la medición de rendimiento de las bases de datos NoSQL documentales versus bases de datos relacionales.

4.1. Definición de ambiente de prueba

Todas las pruebas se realizarán bajo las mismas condiciones de hardware, tiempo, estructuración básica y disponibilidad de recursos, esto con la finalidad de no proporcionar ninguna clase de ventaja o desventaja a cualquiera de las bases de datos sometidas a la prueba.

La prueba en los distintos escenarios se realizará con las siguientes especificaciones técnicas del sistema operativo y software para la simulación de las consultas a las bases de datos por parte de los usuarios.

- Sistema Windows 7 en la versión Ultimate
- Procesador Intel® Core™ i3
- Memoria RAM de 1024 MB
- Disco duro con capacidad de 30 GB

El software Jakarta-Jmeter utilizado en la versión 2,4 para realizar las pruebas de estrés, para las distintas bases de datos, dicha aplicación de software cuenta con la posibilidad de realizar pruebas a cualquier base de datos que se soporte conexiones ODBC.

A continuación se presenta la tabla XI con el resumen de los aspectos y características más importantes de la primera base de datos relacional sometida a las pruebas de rendimiento.

Tabla XI. **Especificaciones técnicas base de datos PostgreSQL**

| Característica | Soporte |
|---------------------------------------|--|
| Versión | 9,1,4 |
| Tipo | Relacional |
| ACID | Si |
| Tamaño máximo de la base de datos | Ilimitado |
| Tamaño máximo de la tabla | 32 TB |
| Tamaño máximo de campo | 1 GB |
| Cantidad máxima de columnas por tabla | 250- 1600 (dependiendo del tipo) |
| Sistemas operativos | Linux y Unix, Mac OS X y Windows 32/64 bit |

Fuente: PostgreSQL. http://www.postgresql.org/es/sobre_postgresql. Consulta: noviembre de 2012.

En la tabla XII se presenta un resumen de los aspectos y características de la primera base de datos NoSQL sometida a las pruebas de rendimiento.

Tabla XII. **Especificaciones técnicas base de datos MongoDB**

| Característica | Soporte |
|--------------------------------------|--|
| Versión | 10gen |
| Tipo | NoSQL documental |
| ACID | No |
| Tamaño máximo de la base de datos | Ilimitado |
| Tamaño máximo del nodo | 3 TB |
| Cantidad máxima de nodos | Ilimitado |
| Tamaño máximo de documentos por nodo | 12 millones |
| Sistemas operativos | Linux y Unix, Mac OS X y Windows 32/64 bit |

Fuente: PostgreSQL. <http://www.10gen.com/use-case/high-volume-data-feeds>. Consulta: noviembre de 2012.

La tabla XIII contiene los aspectos técnicos de la segunda base de datos relacional sometida a las pruebas de rendimiento.

Tabla XIII. **Especificaciones técnicas base de datos MySQL**

| Característica | Soporte |
|---------------------------------------|--|
| Versión | 5,5,24 |
| Tipo | Relacional |
| ACID | Si |
| Tamaño máximo de la base de datos | Ilimitado |
| Tamaño máximo de la tabla | 2 TB |
| Tamaño máximo de campo | 1 GB |
| Cantidad máxima de columnas por tabla | 1- 15(dependiendo del tipo) |
| Sistemas operativos | Linux y Unix, Mac OS X y Windows 32/64 bit |

Fuente: PostgreSQL <http://dev.mysql.com/doc/refman/5.0/es/table-size.html>. Consulta: noviembre de 2012.

Tomando como base las especificaciones del equipo descritas anteriormente se plantean los siguientes escenarios de prueba, para cada una de las bases de datos.

4.1.1. Definición de escenarios de prueba

Las especificaciones de los escenarios de prueba se detallan en la tabla XIV.

Tabla XIV. **Especificaciones de escenarios**

| Especificaciones | Escenario 1 | Escenario 2 |
|-------------------------|--|--|
| Duración | 10 min. | 10 min. |
| Descripción | Simulación de 10 000 usuarios realizando 100 inserciones a la base de datos cada 15 seg. | Simulación de 100 000 usuarios realizando 100 inserciones a la base de datos cada 15 seg. |
| Políticas de inserción | Consultas a una tabla para las bases de datos relacionales y a una colección en el caso de la base de datos NoSQL. | Consultas a una tabla para las bases de datos relacionales y a una colección en el caso de la base de datos NoSQL. |

Fuente: elaboración propia.

La estructura utilizada en todas las bases de datos sometidas a las pruebas de rendimiento se describe en la tabla XV.

Tabla XV. **Estructura de la tabla/colección para los escenarios**

| PUBLICACIÓN |
|--------------------|
| Nombre (char) |
| Nickname (char) |
| Texto (text) |

Fuente: elaboración propia.

4.1.2. **Variables a evaluar**

En la tabla XVI se enumeran las variables internas de salida y entrada de las cuales se tiene control y mediante ellas se realizan los respectivos análisis para determinar el rendimiento de las distintas bases de datos en los escenarios descritos con anterioridad.

Tabla XVI. **Variables para evaluación**

| Tipo de variable | Variables |
|-------------------------|---|
| Internas de entrada | <ul style="list-style-type: none"> • Tiempo entre inserciones • Cantidad de inserciones • Tiempo total de inserciones |
| Internas de salida | <ul style="list-style-type: none"> • Tiempo de respuesta • Consumo de recursos • Porcentaje de fallo • Disponibilidad |

Fuente: elaboración propia.

El análisis de rendimiento se basará únicamente en las variables internas de salida.

A continuación se describirán brevemente las variables que formarán parte del análisis de rendimiento de las distintas bases de datos.

- Tiempo de respuesta: es el tiempo medio que la base de datos tarda en realizar el total de inserciones solicitadas.
- Consumo de recursos: es el porcentaje de uso de CPU, memoria RAM, uso de procesador y espacio en disco utilizado durante las inserciones a la base de datos.
- Porcentaje de fallo: medición del porcentaje de inserciones exitosas y de inserciones fallidas a la base de datos.
- Disponibilidad: porcentaje de accesos exitosos y fallidos a la base de datos.

4.2. Análisis comparativo PostgreSQL, MySQL y MongoDB 10gen

Como se describió en los apartados anteriores, las pruebas realizadas a las bases de datos consisten en la inserción masiva y concurrente de datos durante 10 minutos, a continuación se muestran, describen y analizan los resultados obtenidos en los diversos escenarios.

4.2.1. Tiempos de respuesta

El tiempo de respuesta hace referencia al lapso de tiempo para realizar la inserción de un registro dentro de la base de datos. En la tabla XVII se listan los resultados de inserciones obtenidos en las tres bases de datos para el escenario 1.

Tabla XVII. **Tiempos de inserción escenario No. 1**

| Parámetros | PostgreSQL | MySQL | MongoDB |
|--------------------------------------|-------------------|--------------|----------------|
| Total de accesos | 501 517 | 199 959 | 760 120 |
| Intentos de inserciones cada 15 seg. | 100 | 100 | 100 |
| Tiempo medio por inserción (ms.) | 212 | 1640 | 107 |

Fuente: elaboración propia.

A continuación se presenta la tabla XVIII, con los resultados obtenidos en las pruebas de inserción de las tres bases de datos en el escenario 2.

Tabla XVIII. **Tiempos de inserción escenario No. 2**

| Parámetros | PostgreSQL | MySQL | MongoDB |
|--------------------------------------|-------------------|--------------|----------------|
| Total de accesos | 281 132 | 236 779 | 895 003 |
| Intentos de inserciones cada 15 seg. | 1 000 | 1 000 | 1 000 |
| Tiempo medio por inserción (ms.) | 6 280 | 16 460 | 121 |

Fuente: elaboración propia.

De acuerdo a los resultados obtenidos en las pruebas de los dos escenarios se observa que las bases de datos relacionales insertan menos datos que la base de datos NoSQL, esto se debe a que el tiempo medio de inserción por dato es más elevado en las primeras con respecto a la base de datos NoSQL documental.

4.2.2. Consumo de recursos

Durante el tiempo que se realizaron las pruebas de los distintos escenarios en todo momento se monitorizó el estado de los recursos del computador, en el cual se encontraban las bases de datos, a continuación se muestran los resultados obtenidos correspondientes al consumo de recursos del computador por parte de las bases de datos.

En la tabla XIX se presenta un resumen del porcentaje de recursos consumidos durante el escenario 1 por las bases de datos sometidas a las pruebas.

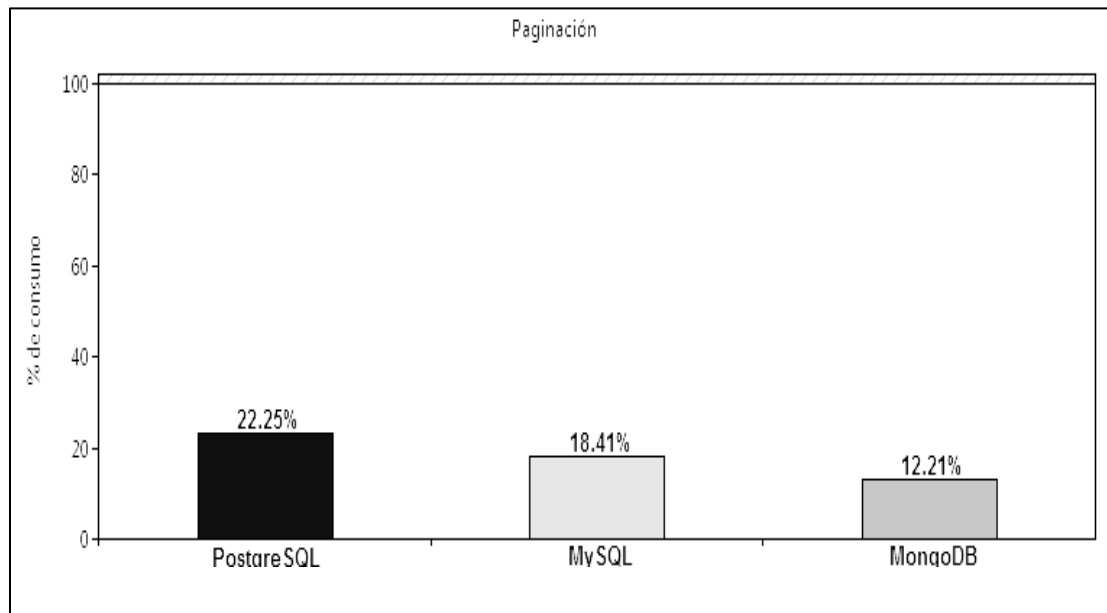
Tabla XIX. **Consumo de recursos escenario No. 1**

| Recurso | PostgreSQL | MySQL | MongoDB |
|-----------------------|-------------------|--------------|----------------|
| Archivo de paginación | 22,25 % | 18,41 % | 12,21 % |
| Disco físico | 20,50 % | 55,13 % | 15,62 % |
| Procesador | 69,64 % | 72,55 % | 23,79 % |
| Memoria RAM | 48,31 % | 43,39 % | 20,92 % |

Fuente: elaboración propia.

Los resultados porcentuales del uso del recurso de paginación por cada una de las bases de datos sometidas a las pruebas del escenario 1 se pueden observar en la figura 5.

Figura 6. **Paginación en escenario No. 1**

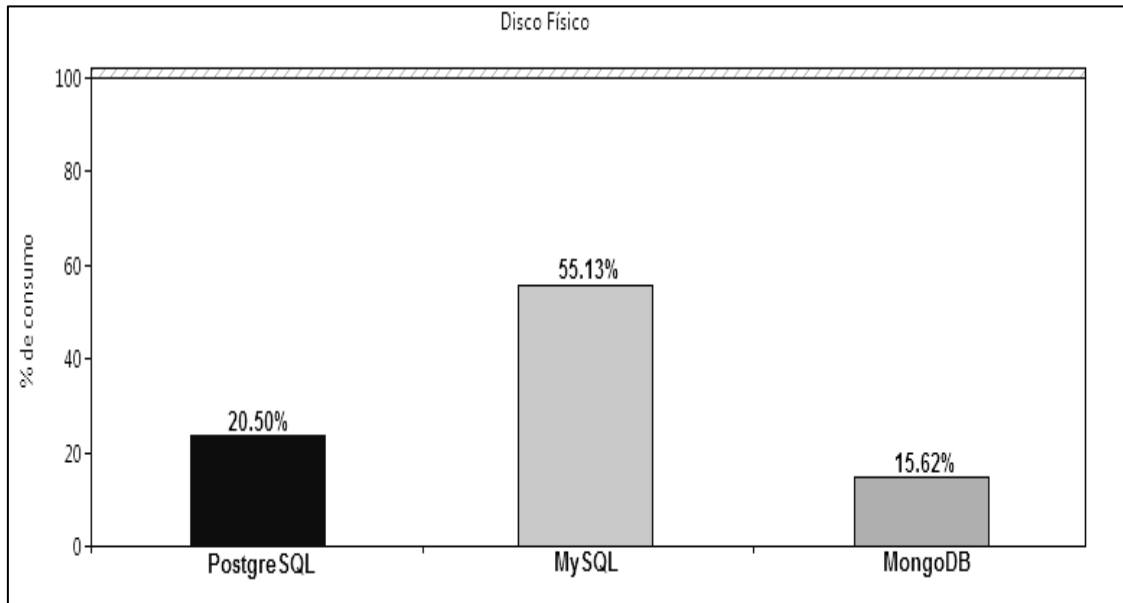


Fuente: elaboración propia.

Como se observa, el uso de paginación por parte de las bases de datos relacionales es relativamente bajo, pero a pesar de eso presentan un consumo mayor con respecto a la base de datos NoSQL documental. Se debe considerar que a mayor porcentaje de consumo de recursos se incurre en un menor rendimiento del equipo.

En la figura 6 se observan los porcentajes de utilización de disco duro o acceso al mismo, por parte de las bases de datos sometidas a las pruebas del escenario 1.

Figura 7. Acceso a disco físico en escenario No. 1

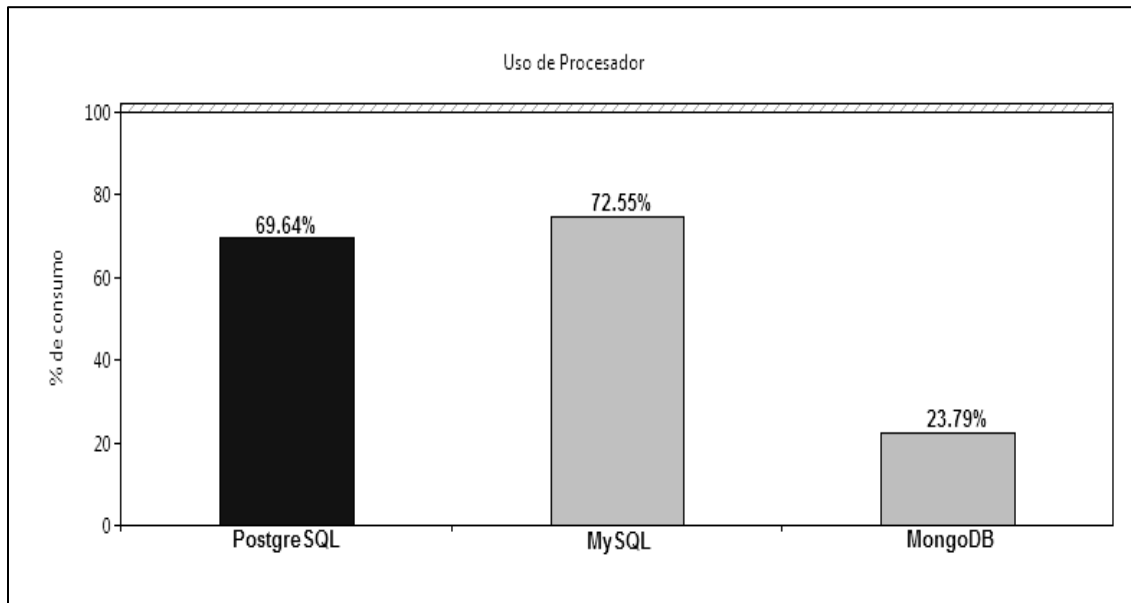


Fuente: elaboración propia.

El mayor porcentaje de acceso a disco físico lo realiza el DBMS MySQL con un 55 %, lo cual representa una carga para el sistema y por consiguiente un bajo rendimiento del equipo.

El porcentaje de uso de procesador que realizó cada una de las bases de datos durante la realización de las pruebas del escenario 1 se encuentra representada de forma gráfica en la figura 7.

Figura 8. **Uso de procesador en escenario No. 1**



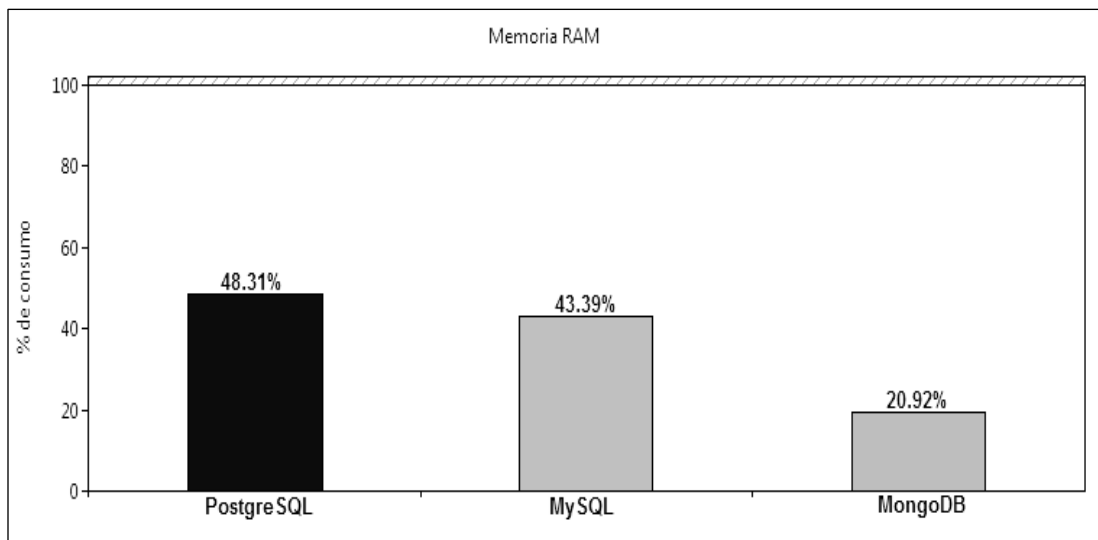
Fuente: elaboración propia.

Uno de los recursos más importantes que afectan el rendimiento del equipo es el consumo de procesador, por consiguiente una aplicación que consume menos recursos de procesamiento es más favorable que una que consume la mayor parte del procesador.

El uso de procesador, por parte de PostgreSQL y de MySQL es notablemente elevado, representando más del 50 % de la capacidad de procesamiento, por lo consiguiente reducen en gran manera el rendimiento del equipo, por otra parte, el consumo de procesador que realiza MongoDB es considerablemente bajo.

A continuación, en la figura 8 se muestra el consumo porcentual de memoria RAM en el cual incurrió cada base de datos durante las pruebas realizadas en el escenario 1.

Figura 9. **Consumo de memoria RAM en escenario No. 1**



Fuente: elaboración propia.

El consumo de memoria RAM, por parte de PostgreSQL, es casi del 50 % del total del equipo, esto posiblemente es causado por el poco uso de archivo de paginación que realiza el programa, de igual manera el uso de memoria que realiza MySQL es considerablemente elevado si se compara con el consumo de memoria RAM por parte de MongoDB.

En la tabla XX se muestra el resumen de resultados obtenidos durante las pruebas realizadas a las tres bases de datos en el escenario 2.

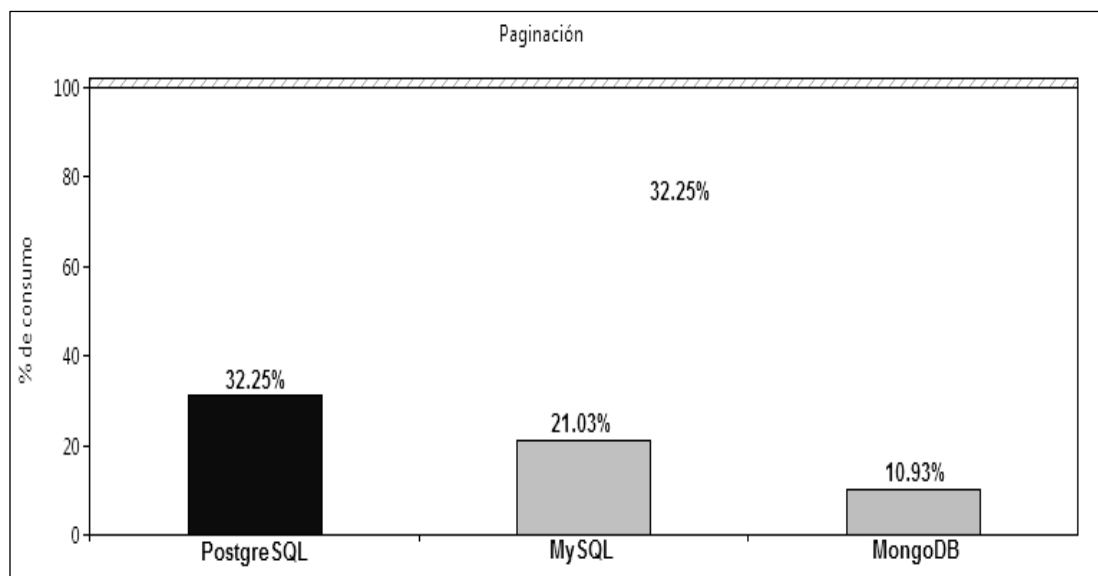
Tabla XX. **Consumo de recursos escenario No. 2**

| Recurso | PostgreSQL | MySQL | MongoDB |
|-----------------------|------------|---------|---------|
| Archivo de Paginación | 32,25 % | 21,03 % | 10,93 % |
| Disco Físico | 34,50 % | 60 % | 49,33 % |
| Procesador | 76,64 % | 82,9 % | 31,67 % |
| Memoria RAM | 56,31 % | 45,63 % | 31,93 % |

Fuente: elaboración propia.

En la figura 9 se observa de manera porcentual la utilización del recurso de paginación, por parte de las tres bases de datos en el escenario 2.

Figura 10. **Paginación en escenario No. 2**

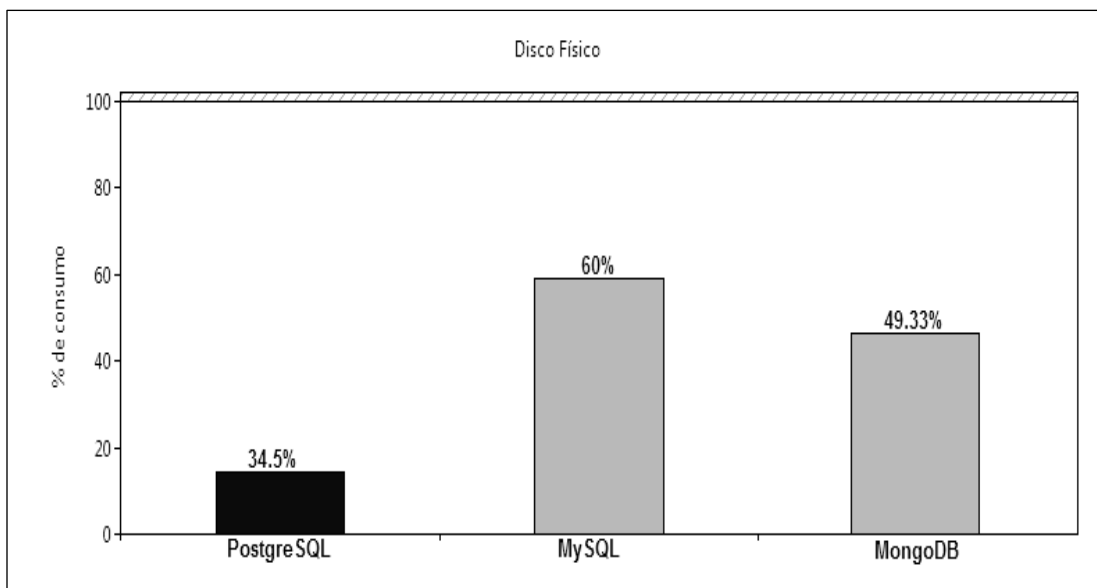


Fuente: elaboración propia.

El uso de archivo de paginación por parte de PostgreSQL en promedio es de 32,25 % el cual es mayor al de los otros dos DBMS, lo cual ayuda a reducir el acceso a disco por parte de la base de datos.

El porcentaje de acceso a disco físico utilizado por las bases de datos durante las pruebas del escenario 2 se encuentra representado en la figura 10.

Figura 11. **Acceso a disco físico en escenario No. 2**

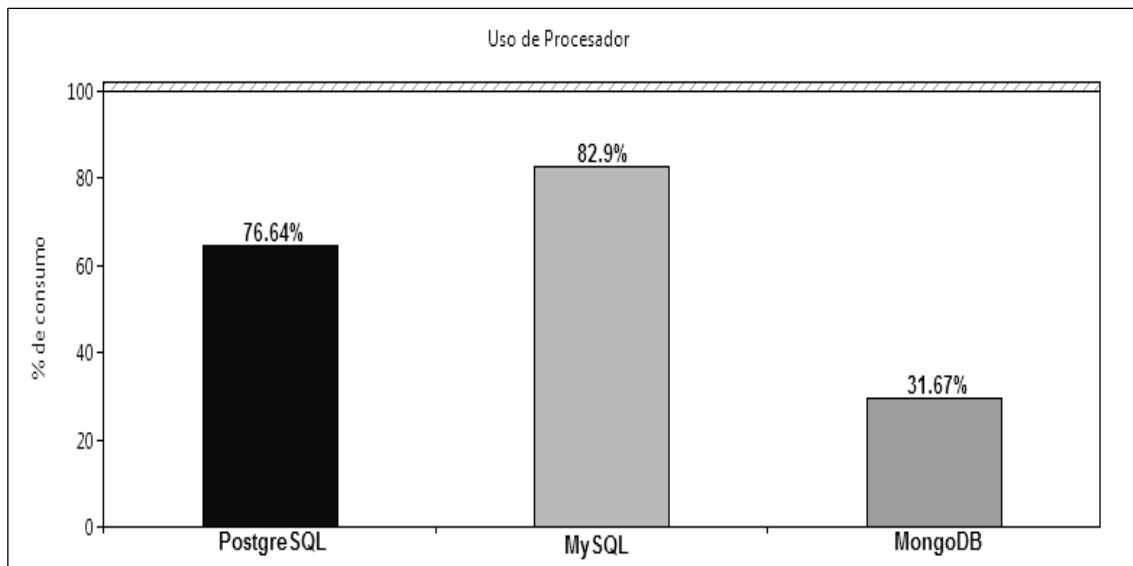


Fuente: elaboración propia.

El porcentaje de acceso a disco en el caso de PostgreSQL es más reducido que el de las otras bases de datos, esto debido a que realizó mayor paginación, por lo tanto no fue necesario acceder a disco para insertar los datos, ya que todo se manejó en memoria. Aunque es importante mencionar que este porcentaje bajo de acceso a disco puede provocar un bajo número de inserciones exitosas a la base de datos.

A continuación se presentan los porcentajes de consumo de procesador por parte de las bases de datos en las pruebas realizadas en el escenario 2, los cuales están representados de forma gráfica en la figura 11.

Figura 12. **Uso de procesador en escenario No. 2**

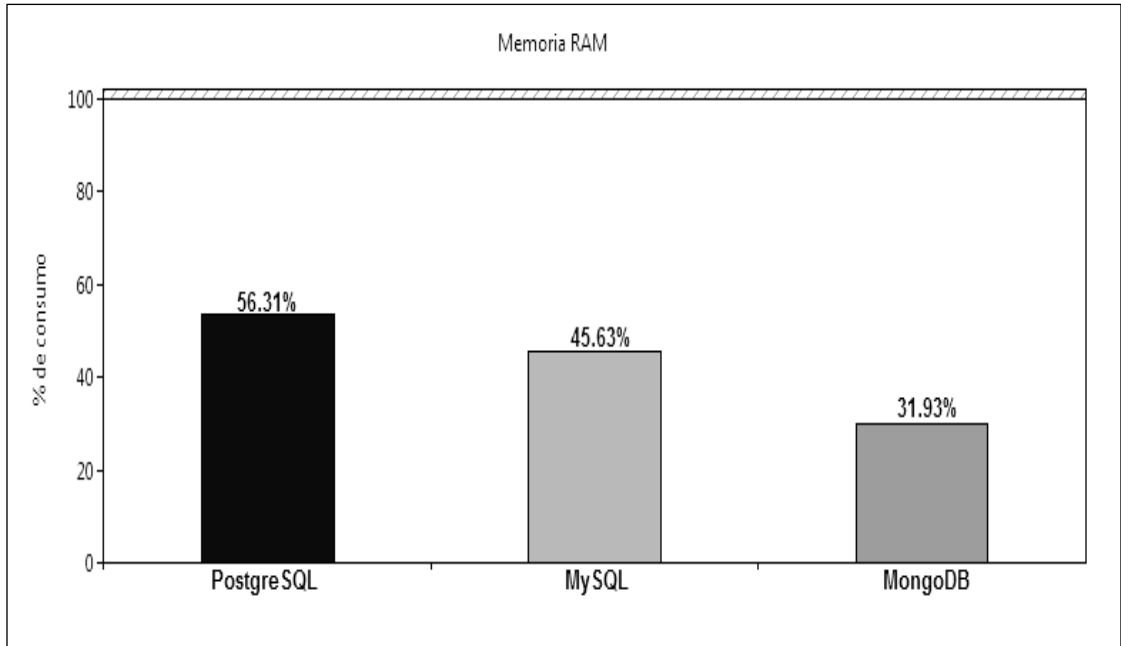


Fuente: elaboración propia.

El uso de procesador por parte de PostgreSQL y MySQL es demasiado elevado, lo cual posee un impacto considerable en el rendimiento del equipo, por el contrario MongoDB realiza un consumo moderado del procesador lo cual ayuda a la mejora del rendimiento.

En la figura 12 se muestra el consumo de memoria RAM en el que incurrieron las bases de datos durante las pruebas realizadas en el escenario 2.

Figura 13. **Consumo de memoria RAM en escenario No. 2**



Fuente: elaboración propia.

El consumo de memoria RAM por parte de PostgreSQL es considerablemente elevado y afecta al rendimiento del equipo.

4.2.3. Respuesta a fallos

Uno de los objetivos primordiales de la realización de las pruebas de rendimiento, para las diversas bases de datos es medir el porcentaje de inserciones fallidas durante toda la prueba, a continuación se muestran los resultados obtenidos por cada base de datos en los diversos escenarios.

Las pruebas de fallos están configuradas con la opción de continuar al momento de encontrar un error, con la finalidad de no interrumpir la prueba.

A continuación se describe mediante una tabla el total de accesos que se tuvo a la base de datos, de los cuales algunos tuvieron éxito al momento de realizar la inserción de los datos solicitados y otros fueron rechazados.

Tabla XXI. **Errores de inserción en escenario No. 1**

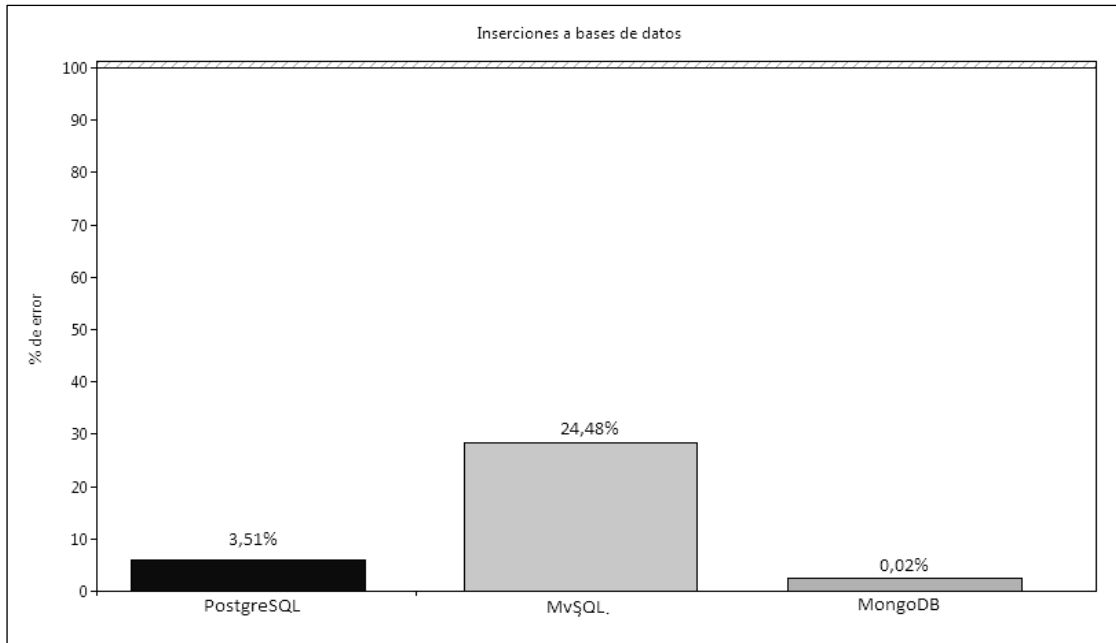
| Parámetros | PostgreSQL | MySQL | MongoDB |
|------------------------------|-------------------|--------------|----------------|
| Total de accesos | 501 517 | 199 959 | 760 120 |
| Duración de la prueba (min.) | 10 | 10 | 10 |
| Inserciones fallidas | 17 603 | 48 950 | 152 |
| Inserciones exitosas | 483 914 | 151 009 | 759 968 |
| Porcentaje de error | 3,51 % | 24,48 % | 0,02 % |

Fuente: elaboración propia.

Como se observa, el porcentaje de error de la base de datos MongoDB es el más bajo, lo cual indica que soporta mayor concurrencia de accesos e inserciones que las otras bases de datos del tipo relacional. A pesar que el porcentaje de errores de las bases de datos relacionales no es tan elevado, la base de datos NoSQL documental posee un porcentaje de error considerablemente menor que el de las relacionales.

En la figura 13 se presenta un resumen del porcentaje de errores en las inserciones para cada una de las bases de datos en las pruebas realizadas en el escenario 1.

Figura 14. **Porcentaje de errores de inserción en escenario No. 1**



Fuente: elaboración propia.

Se observa un bajo porcentaje de inserciones erróneas en la base de datos MongoDB siendo este menor del 1 %, por el contrario MySQL es la base de datos que posee el mayor porcentaje de error lo cual conlleva a inconsistencia en los datos, datos dañados o ilegibles.

En la tabla XXII se presentan los resultados de inserciones exitosas e inserciones rechazadas por cada una de las bases de datos durante las pruebas realizadas en el escenario 2, al final se encuentra el porcentaje que representa la cantidad de errores incurridos por cada base de datos.

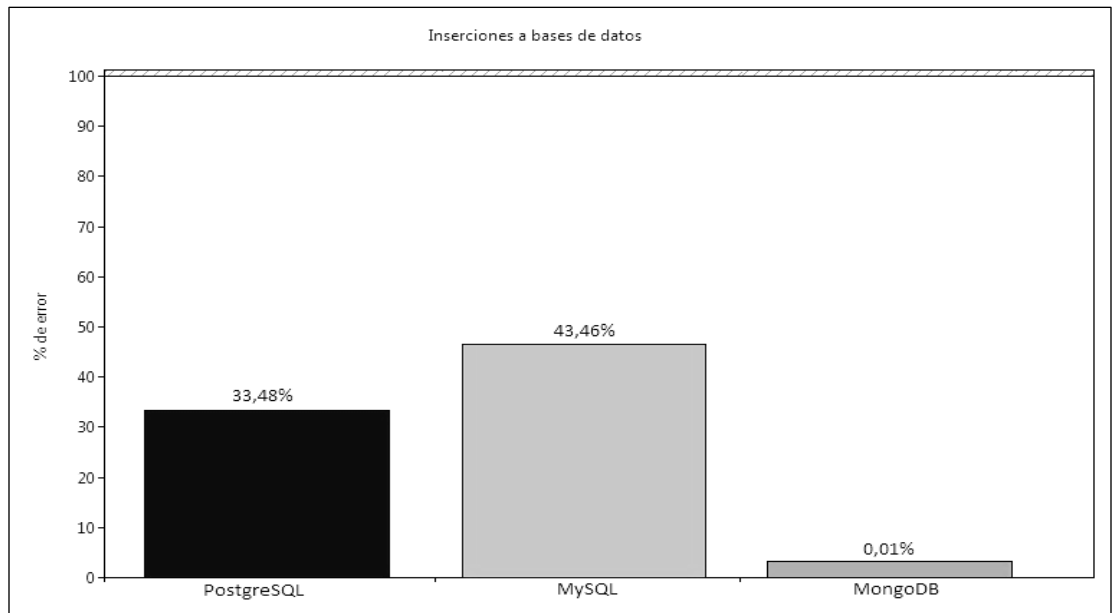
Tabla XXII. Errores de inserción en escenario No. 2

| Parámetros | PostgreSQL | MySQL | MongoDB |
|------------------------------|------------|---------|---------|
| Total de accesos | 281 132 | 236 779 | 895 003 |
| Duración de la prueba (min.) | 10 | 10 | 10 |
| Inserciones fallidas | 94 122 | 102 904 | 89 |
| Inserciones exitosas | 187 010 | 133 875 | 894 914 |
| Porcentaje de error | 33,48 % | 43,46 % | 0,01 % |

Fuente: elaboración propia.

Al igual que en el escenario 1 se observa que el porcentaje de error de la base de datos MongoDB es el más bajo, lo cual indica que soporta mayor concurrencia de accesos e inserciones.

Figura 15. Porcentaje de errores de inserción en escenario No. 2



Fuente: elaboración propia.

4.2.4. Disponibilidad

La medición del grado de disponibilidad se basó en la medición de cuántos usuarios pudieron hacer uso del servicio de la base de datos y realizar el intento de inserción a la base de datos.

No hay que confundir las inserciones fallidas con las fallas de disponibilidad, ya que en el caso de inserciones fallidas el usuario tuvo acceso al servicio y en el caso de fallas de disponibilidad indica que el usuario no pudo si quiera acceder a la base de datos para realizar el intento de inserción.

En la tabla XXIII que se presenta a continuación se resumen los accesos fallidos y exitosos a la base de datos.

Tabla XXIII. **Errores de acceso a base de datos en escenario No. 1**

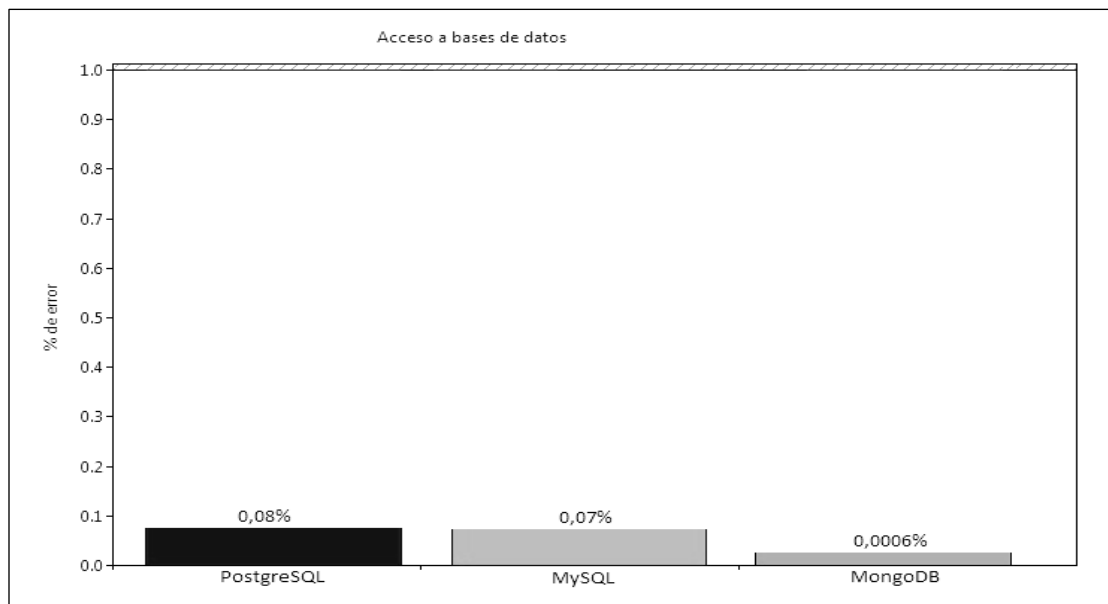
| Parámetros | PostgreSQL | MySQL | MongoDB |
|------------------------------------|------------|---------|----------|
| Total intentos de acceso | 501 929 | 200 100 | 760 125 |
| Duración total de la prueba (min.) | 10 | 10 | 10 |
| Accesos al servicio fallidos | 412 | 141 | 5 |
| Accesos al servicio exitosos | 501 517 | 199 959 | 760 120 |
| Porcentaje de error | 0,08 % | 0,07 % | 0,0006 % |

Fuente: elaboración propia.

Como se observa en la tabla XXIII la base de datos que posee una mejor tasa de accesos es la NoSQL documental, ya que únicamente obtuvo 5 accesos fallidos durante las pruebas realizadas lo cual representa un 0,0006 % de error el cual es considerablemente bajo.

El porcentaje de accesos fallidos o error al acceder a la base de datos en las pruebas del escenario 1 se representa en la figura 15.

Figura 16. **Porcentaje de errores de acceso a base de datos No. 1**



Fuente: elaboración propia.

Las bases de datos de PostgreSQL y MySQL se encuentran muy cercanas en cuanto a errores de acceso se refiere, a pesar de que el porcentaje de error de acceso no es demasiado alto en las bases de datos relacionales, al compararlo con el porcentaje de error que posee la base de datos NoSQL documental MongoDB se aprecia una diferencia significativa.

En la tabla XXIV se resumen los resultados de accesos exitosos y fallidos a cada una de las bases de datos ocurridos durante las pruebas realizadas en el escenario 2.

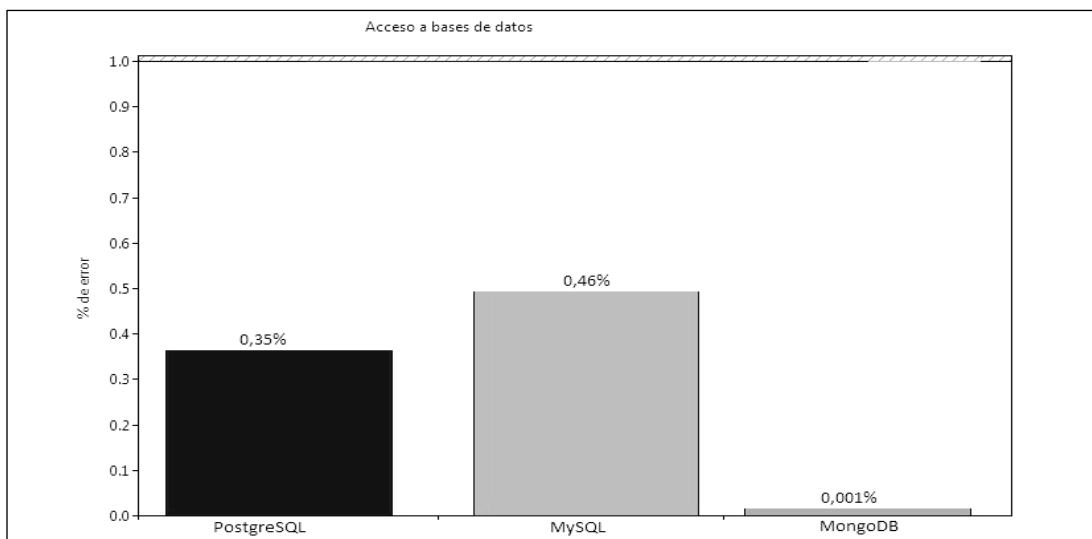
Tabla XXIV. **Errores de acceso a base de datos en escenario No. 2**

| Parámetros | PostgreSQL | MySQL | MongoDB |
|------------------------------------|------------|---------|---------|
| Total intentos de acceso | 282 132 | 237 879 | 895 013 |
| Duración total de la prueba (min.) | 10 | 10 | 10 |
| Accesos al servicio fallidos | 1 000 | 1 100 | 10 |
| Accesos al servicio exitosos | 281 132 | 236 779 | 895 003 |
| Porcentaje de error | 0,35 % | 0,46 % | 0,001 % |

Fuente: elaboración propia.

Debido a que el número de intentos de acceso a las bases de datos aumentó, de igual manera se incrementó el margen de error de accesos en todas las bases de datos, a pesar del aumento como se observa la base de datos NoSQL documental continua poseyendo un porcentaje de error menor del 1 %.

Figura 17. **Porcentaje de errores de acceso a base de datos No. 2**



Fuente: elaboración propia.

Como se observa en la figura 17, las bases de datos relacionales PostgreSQL y MySQL aumentaron considerablemente el margen de error al aumentar el número de usuarios, por el contrario, la base de datos NoSQL documental MongoDB aumentó el margen de error, pero a pesar de esto sigue siendo notablemente menor al de las otras dos.

4.3. Evaluación de resultados

Como se observa en los resultados obtenidos durante las pruebas realizadas a las distintas bases de datos, se concluye que la mejor opción para una solución web que no involucre transacciones monetarias u otra operación de alta importancia que necesite, ACID es una base de datos NoSQL documental.

En la tabla XXV se resumen los aspectos evaluados en las pruebas en ambos escenarios y se indica cuál es el ganador en cada prueba mediante los números 1, 2 y 3, siendo 1 el mejor, 2 el segundo mejor y 3 el peor.

Tabla XXV. **Resumen de pruebas**

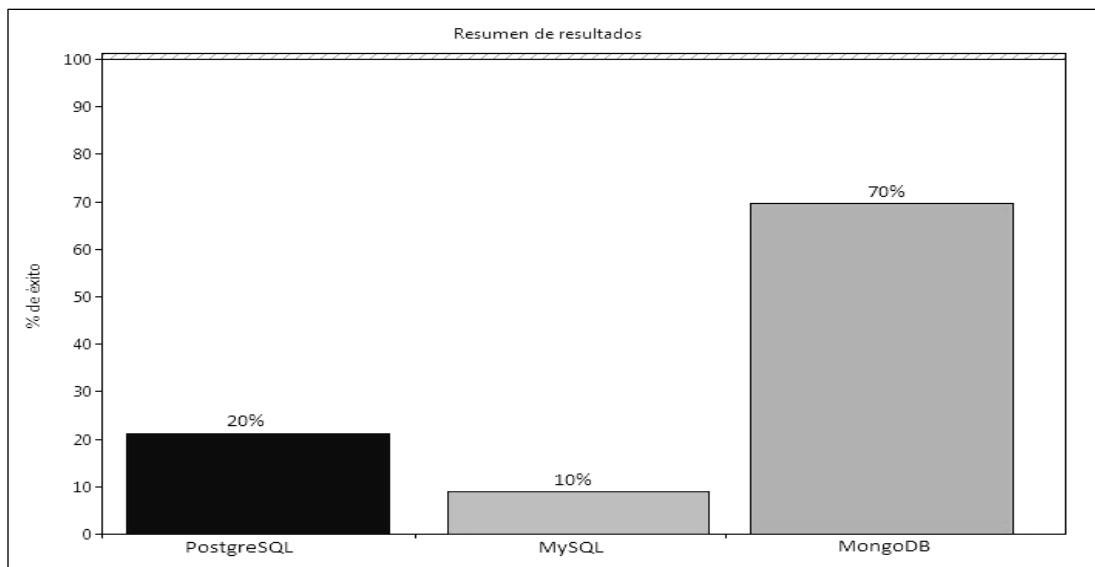
| Prueba | PostgreSQL | MySQL | MongoDB |
|--|-------------------|--------------|----------------|
| Mejor tiempo inserción a la base de datos. | 2 | 3 | 1 |
| Mejor tiempo de acceso a la base de datos. | 3 | 2 | 1 |
| Menor consumo de memoria RAM. | 3 | 2 | 1 |
| Menor consumo de procesador. | 2 | 3 | 1 |
| Mejor uso de paginación de archivos. | 1 | 2 | 3 |
| Menor porcentaje de errores. | 2 | 3 | 1 |

Fuente: elaboración propia.

Con base en los resultados descritos en la tabla XXV, es evidente que la base de datos NoSQL documental MongoDB en la mayoría de aspectos, supera a las bases de datos relacionales de PostgreSQL y MySQL, lo cual la convierte en la mejor opción al momento de implementar una solución para un sitio de web de alto rendimiento.

El porcentaje de éxito de implementar una solución con cada una de las bases de datos evaluadas en los apartados anteriores se puede observar de forma gráfica en la figura 17, la cual se presenta a continuación:

Figura 18. **Porcentaje de éxito en implementación de solución**



Fuente: elaboración propia.

La mejor opción al momento de implementar una solución para el desarrollo de un sitio web de alto rendimiento, es la implementación de una base de datos NoSQL documental, ya que el porcentaje de tener éxito es del 70 por ciento, el cual es notablemente superior al caso en el que se implemente una base de datos relacional.

Los tiempos de respuesta de MongoDB superan por mucho a la mayoría de bases de datos relacionales, es por esta razón que grandes compañías como Disney, MTV, Foursquare, entre otras, han optado por la implementación de esta base de datos en los sitios. La gran facilidad con la que escala horizontalmente es una gran ventaja que ayuda a que MongoDB sea más utilizada cada día.

MongoDB es una base de datos del tipo NoSQL documental, de acuerdo a los resultados obtenidos en las pruebas realizadas en los 2 escenarios que se plantearon anteriormente, se postula como una muy buena opción para reemplazar el uso de las bases de datos relacionales en los sitios web que buscan un alto rendimiento y disponibilidad.

Es importante denotar que no todas son ventajas al momento de implementar MongoDB como base de datos en una aplicación web, ya que posee algunos inconvenientes como: el manejo de ACID y de transacciones.

CONCLUSIONES

1. Es posible el desarrollo de sitios web de alto rendimiento utilizando base de datos NoSQL documentales.
2. La naturaleza de los sitios web que han implementado el uso de las bases de datos NoSQL documentales y que han obtenido beneficios sustanciales son la administración de contenido, entretenimiento, redes sociales o para mostrar información en tiempo real.
3. La base de datos NoSQL documental más utilizada en la actualidad, por las compañías debido a las múltiples ventajas, como la funcionalidad intuitiva, es MongoDB 10gen.
4. El grado de complejidad en la implementación de una base de datos NoSQL documental en un sitio web, es relativa a los conocimientos de programación que posea el personal encargado de integrarla al sitio.
5. Las bases de datos NoSQL documentales, como es el caso de MongoDB 10gen, no poseen seguridad o integridad en los datos, es por esta razón que no se deben utilizar en transacciones importantes, por ejemplo, transacciones bancarias.
6. La implementación de MongoDB 10gen supone ahorro en gastos de infraestructura para la empresa.

RECOMENDACIONES

1. Realizar un análisis de viabilidad con anterioridad a la puesta en marcha de la integración de dicha tecnología, con la finalidad de no tener ningún inconveniente.
2. Adquirir más información acerca del tema de las bases de datos NoSQL documentales y de la gran variedad que se encuentran en el mercado, como: CouchDB, RavenDB, SisoDB, por mencionar algunas, ya que MongoDB no es la única opción en cuanto a este tema.
3. Capacitar al personal que no posea mucha experiencia en programación, esto con la finalidad de hacer más agradable la transición a la nueva tecnología.
4. Analizar la naturaleza de la empresa o lógica de negocio para verificar qué tipo de base de datos NoSQL es la más apta para la solución de las problemáticas que se posean.

BIBLIOGRAFÍA

1. A.S. FOUNDATION. *Cassandra* [en línea]. [ref. de enero 2009].
Disponible en Web: <<http://cassandra.apache.org/>>.
2. KADUX. *Base de datos noSQL mongoDB* [en línea]. [ref. de abril 2011].
Disponible en Web: <<http://www.ubuntu-es.org/node/153667>>.
3. MONGODB. *MongoDB 10gen Customers* [en línea]. [ref. de abril 2009].
Disponible en Web: <<http://www.10gen.com/customers/>>.
4. *NoSQL* [en línea]. [ref. de agosto 2009]. Disponible en Web:
<<http://nosql-database.org/>>.
5. OSORIO RIVERA, Fray Leon. *Bases de datos relacionales teoría y práctica*. Medellín: Textos Académicos, 2008. 249 p.
6. PALUMBO, Gianfranco. *Schema Desing* [en línea]. [ref. de abril 2009].
Disponible en Web:
<<http://www.mongodb.org/display/DOCS/Schema+Design> >.

ANEXOS

Anexo 1. Configuración de Jmeter para realización de pruebas

REQUERIMIENTOS

La aplicación de Jmeter es compatible en cualquier sistema operativo basado en Unix, Windows, Solaris o Mac OS.

CONSTRUCCIÓN DE UN PLAN DE PRUEBA

Una vez descomprimido el archivo .zip, se debe ingresar a la carpeta bin donde se encontrará un archivo llamado jmeter.jar, el cual se debe ejecutar, una vez ejecutado se deben realizar los siguientes pasos.

1. En el menú del lado izquierdo se encuentra una opción llamada plan de pruebas, al cual se debe dar clic secundario, luego las opciones: Añadir->Grupo de hilos.
2. Se desplegará una ventana de configuración.
3. Colocar los siguientes valores: nombre que se le desea dar al grupo de hilos, número de hilos, la cantidad de usuarios que se desea simular, período, el lapso de tiempo entre cada grupo de peticiones.

Fuente: <http://www.oracle.com/technetwork/java/javase/downloads/jre-6u26-download-400751.html>. Consulta: noviembre de 2012.

Anexo 2. Creación de conexión JDBC

1. Una vez finalizadas las configuraciones para el grupo de usuarios se procede a configurar las conexiones JDBC.
2. Se debe seleccionar el nuevo grupo de hilos recién creado y dar clic secundario, luego seleccionar las opciones:
Añadir -> Elemento de configuración -> Configuración de conexiones JDBC.
3. Se desplegará una nueva pantalla en la cual se deberán poner los datos de la conexión JDBC, para esto ya se debe poseer los conectores JDBC que se utilizarán, para el caso de estudio realizado se utilizaron los conectores de Postgresql, MySQL y MongoDB.
4. Se deberán colocar los conectores dentro de la carpeta lib de jmeter.
5. Los datos de la conexión JDBC que se deben llenar son los siguientes: nombre de la conexión, dirección desde la cual se conecta la base de datos, nombre del conector JDBC para cada base de datos, usuario y contraseña, esto si es que existiera.

Fuente: <http://www.mysql.com/downloads/connector/j/>. Consulta: noviembre de 2012.

Anexo 3. Creación de petición JDBC

1. Una vez configurado el JDBC se procede a dar clic secundario sobre el grupo de hilos y se seleccionan las siguientes opciones:
Añadir -> Emisor -> Solicitud JDBC.
2. Se desplegará una pantalla en la cual se deben llenar los campos: nombre que se desee colocar a la prueba, nombre de la variable asignada en la configuración del conector JDBC, la consulta de datos, inserción, actualización o cualquier operación que se desee realizar, con la condición que sea con la estructura que maneja la base de datos a estresar.

Fuente: <http://jmeter.apache.org/>. Consulta: noviembre de 2012.