

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

Comparativo de rendimiento transaccional entre el motor de bases de datos relacional MySQL y el motor de bases de datos NoSQL tipo documental MongoDB.

Cristian Henry González López
Marlon Montoya Torres

Ingeniería de Sistemas

Jorge Iván Bedoya Restrepo

INSTITUTO TECNOLÓGICO METROPOLITANO
Medellín
08/08/2016

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

El crecimiento en el número de datos generados a cada segundo y que cumplen un ciclo o función dentro de los procesos en los que se encuentran los sistemas de información, requiere que el diseño y construcción de software tenga como primordial objetivo la mitigación de riesgos de pérdida de datos y su control absoluto desde el momento en que ingresan a un sistema de información para su procesamiento hasta el punto de almacenamiento en repositorios centralizados para futuros análisis, suponiendo un enorme reto a la sociedad moderna interconectada y generadora de datos.

Campos como la ingeniería de software, que incluye las metodologías de programación de aplicaciones y los modelos de almacenamiento de datos (motores de bases de datos) para la construcción de sistemas de información, han evolucionado con miras a lograr la mejor solución de acuerdo a las características y el contexto de un problema tratado.

El desarrollo de este proyecto tiene como finalidad el análisis comparativo de dos tecnologías de bases de datos, una basada en el modelo relacional y otra en el modelo NoSQL. El análisis se basa en la definición inicial de parámetros de medición tales como rendimiento, confiabilidad, disponibilidad e integridad a partir del desarrollo de un caso práctico donde se almacenará información de contacto de un grupo de personas.

Si bien los resultados nos indican que NoSQL como motor de bases de datos es una solución adecuada en aquellos ambientes donde se requiera velocidad y gran intensidad de carga de datos, no podemos reemplazar aún los modelos relacionales ya que nos siguen brindando integridad en aquellos ambientes de desarrollo donde el paradigma NoSQL carece de la madurez que se requiere.

Palabras clave:

NOSQL: conjunto de almacenes de datos no relacionales

MYSQL: sistema de administración de bases de datos relacional

MAPREDUCE: Modelo de programación que da soporte a la computación paralela sobre grandes colecciones de datos.

JAVA: Lenguaje de programación de propósito general, concurrente, orientado a objetos.

NETBEANS: Entorno de desarrollo IDE integrado libre principalmente para el lenguaje de programación Java.

PHPMYADMIN: Es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario.

APACHE: Es un servidor web HTTP de código abierto multiplataforma

PHP: Lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

En primer lugar agradecer a Dios por brindarnos la oportunidad de desarrollar y poner en práctica nuestras habilidades tanto académicas como humanas en pro de este objetivo profesional.

A nuestro asesor de trabajo de grado asignado por la institución Jorge Iván Bedoya agradecemos por sus claras y oportunas recomendaciones en el proceso tanto de concepción como de desarrollo de tan importante proyecto.

Agradecemos a nuestros padres, hermanos y demás familiares que a través de su ejemplo y acompañamiento constante lograron inculcar en nosotros el tesón necesario para alcanzar la meta trazada.

Por último queremos agradecer a nuestros amigos, compañeros de trabajo, docentes y colegas profesionales que generosamente aportaron sus conocimientos y consejos en el momento que les fue solicitado y que ayudaron a consolidar durante todo el ciclo universitario las bases fundamentales de nuestro desarrollo profesional y humano.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

RDBMS: Sistema de gestión de bases de datos relacionales

ACID: Atomicity, Consistency, Isolation and Durability

CAP: Consistency, Availability, Partition Tolerance

IEEE: The Institute of Electrical and Electronics Engineers

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

Comparativo de rendimiento transaccional entre el motor de bases de datos relacional MySQL y el motor de bases de datos NoSQL tipo documental MongoDB.....

1. INTRODUCCIÓN.....
2. MARCO TEÓRICO.....
3. METODOLOGÍA
4. RESULTADOS Y DISCUSIÓN
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

REFERENCIAS

APÉNDICE.....

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

La evolución tecnológica ha facilitado y ampliado las herramientas empleadas para comunicarnos así como la velocidad con que obtenemos y analizamos cada parte del mundo que nos rodea derivando en un crecimiento exponencial de datos con orígenes tradicionales como los organismos del sector público y privado, centros de investigación científica e instituciones educativas y entes poco comunes décadas atrás como el ciudadano de a pie, los procesos automatizados y de inteligencia artificial.

Los sistemas de información ubicados en cada contexto deben ser diseñados y construidos con el objetivo primordial de mitigar los riesgos existentes, es decir, garantizar el control absoluto como son el procesamiento eficiente y eficaz, el análisis detallado, la perfecta clasificación y el posterior almacenamiento en repositorios confiables para futuras revisiones. Cada dato generado tiene una función acorde al contexto en que se encuentra y la errónea clasificación o pérdida de alguno de ellos puede dotar de una definición diferente a ese entorno.

Hoy en día el modelo relacional se enfrenta a un fenómeno que sugiere un cambio en el almacenamiento y administración de los datos. Este fenómeno ha sacado a flote una serie de deficiencias que si bien en el momento de su establecimiento no se dilucidaron en un futuro cercano, ahora con el crecimiento de los datos, la velocidad de procesamiento y la variedad de los mismos pasando de ser estructurados a desestructurados han cambiado esta tendencia y se ratifican con las limitantes que tiene el modelo para manipular datos geográficos, multimedia, CAD, entre otros.

Stonebraker, M, (2007), menciona que , “los modelos relacionales tienen una arquitectura de más de 25 años cuando las características de Hardware eran muy diferentes a las de hoy y donde el único mercado objetivo era el procesamiento de datos empresariales , en los últimos 25 años han evolucionado otros tipos de mercados con necesidades muy diferentes...” , y es aquí donde el Teorema CAP (consistencia, disponibilidad y tolerancia a partición) representa una verdadera alternativa aplicada a la construcción de entornos distribuidos donde los Sistemas de Gestión de Base de Datos (DBMS) se convierten en una tarea compleja y difícil de alcanzar.

De acuerdo a las nuevas tendencias e intereses de los distintos sectores de la industria obligadas a acudir a tecnologías que les permita agilizar sus procesos de negocio se realiza este comparativo de rendimiento transaccional entre el motor de bases de datos relacional MySQL y el motor de bases de datos NoSQL tipo documental MongoDB, siendo estos motores los más sobresalientes y reconocidos dentro de las tecnologías y comunidades de software libre.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

OBJETIVOS

General

Comparar el rendimiento de los motores de bases de datos MySQL y MongoDB, con el fin de identificar las ventajas transaccionales y de sistema entre los modelos relacional y NoSQL.

Específicos

- Realizar una caracterización del motor de bases de datos relacional MySQL y el motor de bases de datos NoSQL tipo documental MongoDB para sistemas transaccionales y los antecedentes en cuanto su comparación.
- Establecer métricas y pruebas que permitan identificar el rendimiento entre los motores de bases de datos mencionados anteriormente.
- Diseñar y desarrollar un software para la gestión de información de contacto de un grupo de personas que permita validar el modelo relacional y NoSQL basado en las métricas establecidas.
- Evaluar los resultados de la comparación entre los modelos relacional y NoSQL

Se aborda este trabajo de investigación desde la necesidad y curiosidad como estudiante y profesional de la ciencia de ingeniería de sistemas de comparar y conocer los resultados de los rendimientos transaccionales entre dos motores de bases de datos totalmente diferentes desde su concepción hasta su funcionamiento en entornos operativos.

Desde un marco teórico se identificará las principales características de ambos modelos, al igual que se dará a conocer los motores de bases de datos que representan los modelos previamente mencionados y que serán los utilizados para las pruebas de rendimiento establecidas. Tan importante como conocer sus orígenes, enfoques operativos, y generalidades, lo es el conocer los trabajos similares que se han realizado, por lo tanto pondremos en contexto el estado del arte que nos servirá de base para alcanzar nuestro objetivo general.

En el apartado metodológico se describe las fases en las que se desarrolló el proyecto, al igual que las respectivas pruebas de rendimiento que se definieron para determinar el rendimiento de los motores de bases de datos complementándolo con la documentación del aplicativo de software desde el cual se ejecutaron las pruebas.

Por último con la obtención de los resultados habremos cumplido con los objetivos trazados al inicio y con la realización de las respectivas conclusiones y recomendaciones tendremos la materia prima para futuros trabajos de investigación.

2. MARCO TEÓRICO

En esta sección se presentan los conceptos fundamentales que caracterizan la presentación de este proyecto. En primer lugar se hace una descripción de los diferentes motores de bases de datos tanto relacional como no relacional identificando la forma como estos llevan a cabo las operaciones de inserción, eliminación, actualización y consulta, luego se presenta una caracterización del motor de bases de datos relacional MySQL y el motor de bases de datos NoSQL tipo documental MongoDB. Posteriormente se describe el estado de arte en cuanto a la comparación de estos motores incluyendo sus limitaciones.

Bases de datos relacionales

Los sistemas de gestión de bases de datos relacionales (RDBMS) son un estándar en la actualidad para muchas industrias, existen sistemas de gestión propietarios como lo son: Oracle, Microsoft SQL Server y DB2 y de código abierto como MySQL y Postgress. A continuación veremos algunas de sus características principales:

- Oracle:** Como se puede observar en la figura 1, una instancia en Oracle la constituyen un espacio en memoria y una serie de procesos en segundo plano (background), la base de datos de Oracle están basadas en un conjunto de archivos físicos llamados: Control Files, data files y redo logs (figura 2), por su parte las transacciones de la base de datos se registran en un (redo log), por consiguiente no existe disposición para crear archivos de registro individuales para espacios de tablas.

El nivel más alto en la estructura de almacenamiento de datos en Oracle es la base de datos. Una base de datos agrupa piezas lógicas de mayor nivel de almacenamiento llamadas Tablespaces, los Tablespaces pueden estar formados por varios datafiles.

Figura 1: Instancia Oracle

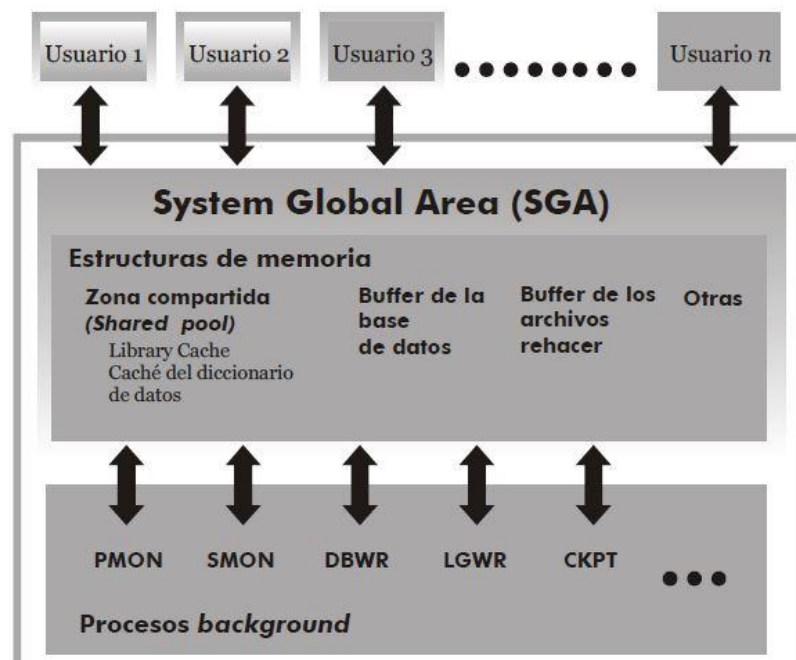
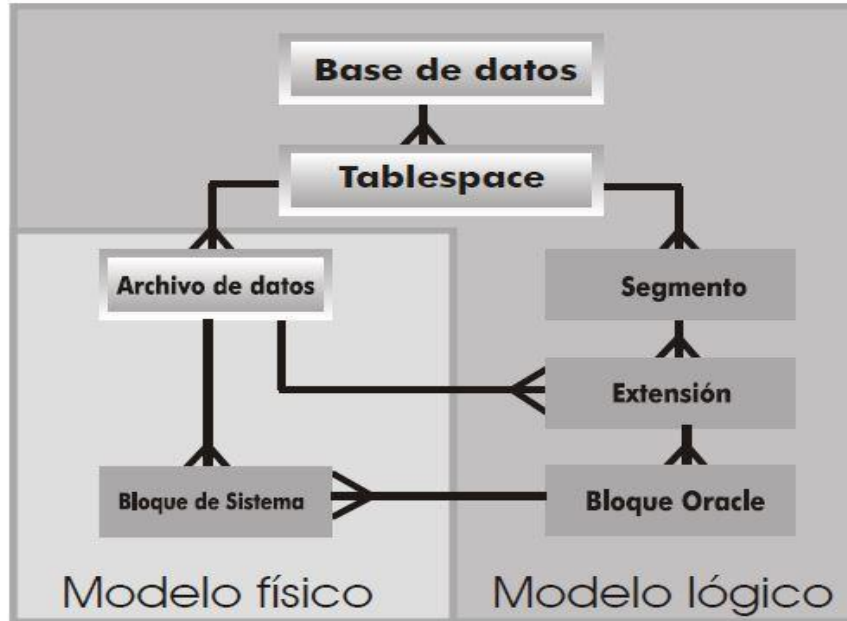


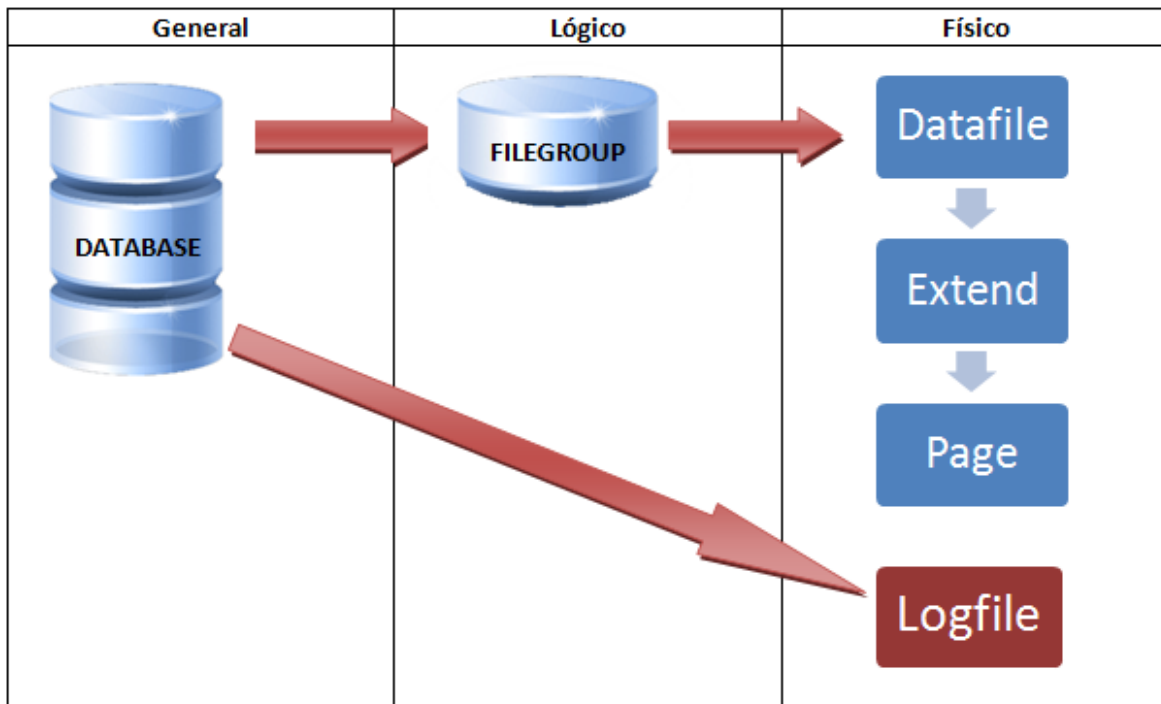
Figura 2: Arquitectura base de datos Oracle



Los cambios en la base de datos no son guardados hasta que tras una serie de instrucciones se decide llevar a cabo esos cambios. Hasta ese momento todo lo realizado se toma como provisional. Un fallo en la máquina permitiría invertir los cambios. Una transacción son varias operaciones SQL que forman una unidad de trabajo. Comienza cuando una persona se conecta y de ahí hasta que ejecuta la instrucción commit (ejecutar la transacción) o rollback (anular la transacción). La anulación deja la base de datos en el estado anterior al comienzo de la transacción. Tras un commit o un rollback comienza la siguiente transacción. En Oracle se admite además el uso de puntos de ruptura (checkpoints) para almacenar valores intermedios y volver a cualquier de ellos si interesa. Pero esto ralentiza excesivamente el sistema.

- SQL Server:** La figura 3 nos muestra cómo está constituida la arquitectura de una base de datos en SQL Server, el concepto de instancia significa un servicio de aplicación auto contenida que implica archivos del sistema operativo, las estructuras de memoria, los procesos de segundo plano y la información de registro. Una instancia está representada por un servicio en Windows y puede estar en ejecución o detenido. La parte fundamental de una instancia en SQL Server son sus bases de datos. Los archivos de datos pueden ser lógicamente agrupados en grupos de archivos. Cada base de datos tendrá su propio registro de transacciones y las propiedades del archivo de registro deberá ser especificado durante la creación de la base de datos.

Figura 3: Arquitectura base de datos SQL Server



Desde el punto de vista lógico, la base de datos debe tener al menos 1 "FileGroup" el cual contiene a toda la metadata de la misma base de datos, es decir tablas y vistas de sistema. Uno de los propósitos de los "FileGroups" es poder distribuir la data a través de varios discos duros físicos, de esta manera se puede obtener mayor rendimiento en las operaciones de I/O debido a que más de un disco trabajara al mismo tiempo. Los "FileGroups" pueden contener 1 o más "Datafiles", y cada uno de estos datafiles se puede encontrar en un discos diferentes, lo cual también agilizará las consultas y los ingresos de información a las tablas que se encuentren asignadas a este "FileGroup".

Bases de datos No relacionales (NoSQL)

La familia de bases de datos NoSQL cobija varios grupos enfocados en un modelo o esquema de solución, entre los que se pueden encontrar:

- Documentales:** bases de datos diseñadas sobre el concepto de "Documento", el cual se refiere a que cada documento encapsula y codifica datos siguiendo un patrón de diseño estándar, no queriendo decir que cada documento es igual a otro siendo estos del mismo tipo. Las escrituras son atómicas a nivel de documento, un modelo de datos no normalizado facilita las operaciones de escritura atómicas. Cada vez que un documento se actualiza la modificación realiza cambios a todos los atributos asociados, cada documento tiene un tamaño máximo de 16 MB, a través de los índices en el campo `_id` se pueden realizar consultas más eficientes.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En las consultas se especifican los criterios o condiciones que identifican los documentos, es posible imponer límites, saltos y criterios de ordenación. Para la actualización y operaciones de borrado, es posible especificar los criterios para seleccionar los documentos para actualizar o eliminar. MongoDB, CouchDB y RavenDB son los principales representantes de esta familia.

- **Grafo:** son bases de datos en las que la información se encuentra representada en forma de grafo y las relaciones con sus propias aristas, permitiendo recorrerlo aplicando la teoría de grafos. Según Gracia del Busto, H. (2012) una base de datos orientada a grafos debe estar absolutamente normalizada, esto quiere decir que cada tabla tendría una sola columna y cada relación tan solo dos nodos. Son apropiadas para gestionar redes sociales, transporte público, mapas de carreteras, etc. Los elementos básicos para un modelamiento en un grafo sería: El nodo (Vértice), la relación (arista) y la propiedad (atributo).

Cada nodo contiene directamente y físicamente una lista de relación de registros que representan sus relaciones con otros nodos. Cada vez que se ejecuta el equivalente de una operación JOIN, la base de datos solo utiliza esta lista y tiene acceso directo a los nodos conectados, eliminando la necesidad de un costoso cálculo de búsqueda. Neo4j, GraphBase son algunos de los motores más representativos de este tipo.

- **Clave/Valor:** En Gracia del Busto, H. (2012) se describe que las bases de datos Clave/Valor son aquellas que cuentan con entidad de almacenamiento el concepto de cabinets en donde son almacenadas tantas parejas clave/valor como sea necesario. Algunos sistemas permiten duplicidad de claves pero otros no. Los sistemas de clave-valor tienen un rendimiento excelente para volúmenes de datos muy grandes, a cambio de ser muy simples y renunciar a funcionalidades que se tienen en otros sistemas como la verificación intrínseca de la integridad de datos. Las validaciones de los datos se delegan completamente en la aplicación cliente, siendo la base de datos, simplemente el lugar donde se guardan los datos.

La inserción y recuperación de los datos se hace a partir de la clave, la consistencia es aplicable sólo para operaciones sobre una única clave, es decir, operaciones como get, put o delete de una clave. Todas las bases de datos de este tipo realizan consultas sobre la clave, siendo muy difícil hacer consultas por alguno de los campos asociados a una clave, es la aplicación la que se tiene que encargar de leer el valor y utilizar el atributo apropiado como elemento de filtrado de consulta, los motores más utilizados de este tipo son: Redis, Riak, Couchbase Server.

3.2.3 Base de datos relacional MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS por sus siglas en inglés) que compite con otros sistemas RDBMS conocidos, como Oracle, SQL Server, DB2, PostgreSQL y MySQL.

Codd, E.F. (1970), publicó un trabajo donde propone un nuevo modelo de datos, llamado desde entonces modelo relacional, siendo el más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente, surge como una alternativa la cual estaba enfocada a la

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

búsqueda de nuevas formas de organizar y acceder a los datos generados. Es precisamente esta organización la que permitió definir una estructura basada en un conjunto de registros almacenados en tablas y relaciones entre estas que permitan el establecimiento de un vínculo o correspondencia.

La independencia física se convierte en uno de los objetivos principales del modelo relacional que indica que la forma de almacenar los datos, no debe influir en su manipulación lógica, esto acompañado de una independencia lógica donde las aplicaciones que utilizan las bases de datos no deber ser modificadas por el hecho que se modifiquen elementos de estas.

Las instrucciones de inserción, actualización y eliminación ejecutadas sobre los datos en una base de datos relacional se denominan transacciones, para que sean denominadas como tal, estas transacciones deben cumplir un conjunto de características que garantizan un comportamiento fiable, este grupo de características se agrupan bajo el acrónimo ACID de Atomicity, Consistency, Isolation, Durability: Atomicidad, consistencia y Durabilidad traducido al español, cada una de estas propiedades se definen a continuación:

- **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.
- **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.
- **Aislamiento:** 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.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

3.2.4 Base de datos tipo documental MongoDB

MongoDB es un motor de bases de datos que se encuentra dentro del grupo de los llamados motores NoSQL. Esta denominación se debe a que dentro de sus principales características se encuentra el no manejar la misma lógica o razonamiento al momento de diseñar, organizar y acceder a los datos como lo hacen los motores relacionales.

NoSQL se refiere a un conjunto de almacenes de datos no relacionales que, si bien no proporcionan las garantías de **Atomicidad, Consistencia, Aislamiento y Durabilidad ACID ISO/IEC 10026-1**, características necesarias para que una instrucción sea considerada como una transacción, Jacob, A.L (2010), se refiere a NoSQL no como un sistema en particular sino más bien

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

a un paso que se da en el modelo relacional (en comparación con el lenguaje de consulta), este proporciona:

- Soporte para grandes volúmenes de datos.
- Altas cargas transaccionales en tiempo real.
- No degradación del rendimiento por los dos ítems anteriores.
- Disponibilidad constante debido a su crecimiento vertical, es decir, facilidad de agregar nuevos nodos (servidores) formando clúster de bases de datos NoSQL con facilidad.
- El tipo de repositorio de datos a modelar e implementar no requiere de consistencia por no ser un factor crítico. Por esto no requiere de las transacciones ACID, logrando además aumentar la disponibilidad, escalabilidad y bajos tiempos de respuesta, es decir, se compensa la pérdida de flexibilidad en tiempo de ejecución con mejoras significativas en escalabilidad y rendimiento al tratar con cierto tipo de datos.
- Facilidad de integración con una aplicación.
- Facilidad de uso y administración.
- No poseen esquemas fijos de tablas ni soportan las instrucciones JOIN normales de los motores de bases de datos que soportan el lenguaje SQL.

Mientras que en el modelo relacional existen un conjunto de propiedades ACID que permiten garantizar que las transacciones ejecutadas sobre los datos se realicen de forma confiable, en la actualidad los altos volúmenes de datos generados llevan a que algunas de estas propiedades se pierdan y la gestión de estos resulte imposible.

Los sistemas basados en el modelo NoSQL deben cumplir con ciertas propiedades que permiten potencializar los sistemas distribuidos a través del cumplimiento combinado de las mismas. Para tales propiedades existe el teorema CAP (Consistency, Availability, Partition Tolerance) que define que un sistema distribuido no puede satisfacer de manera completa los siguientes atributos al mismo tiempo: Consistencia, Alta disponibilidad y Tolerancia a particionamiento. Estas propiedades se definen a continuación:

- **Consistencia:** Un sistema es consistente si todos los nodos de un sistema visualizan la misma información en el mismo instante de tiempo, incluso durante las actualizaciones de los datos.
- **Alta disponibilidad:** Se busca que cualquier petición recibida en un nodo del sistema debe obtener una respuesta que indique si fue exitoso o fallido.
- **Tolerancia a particionamiento:** La tolerancia a particiones se refiere al hecho de garantizar que el sistema continúe en funcionamiento incluso si se pierde de forma arbitraria mensajes entre los nodos del sistema, por ejemplo si se pierde conectividad o se despliega en diferentes servidores lo cual debe ser transparente para el cliente.

“Para cualquier sistema distribuido, usted sólo puede cumplir con dos de estos atributos” (Profesor Eric A. Brewer, 2000), El teorema CAP nos indica que un sistema distribuido solo puede cumplir con dos de estos atributos lo que significa que: el sistema ejecutará las operaciones de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

forma consistente (**Consistencia**), aunque se pierda la comunicación entre nodos (**Tolerancia a particiones**), pero no se asegura que el sistema responda (**Disponibilidad**). Por otro lado el siempre responderá a las peticiones (**Disponibilidad**), aunque se pierda la comunicación entre los nodos (**Tolerancia a particiones**) sin embargo los datos pueden no ser consistentes (**Consistencia**). Por ultimo nos encontramos con que el sistema siempre responderá a las peticiones (**Disponibilidad**) y los datos procesados serán consistentes (**Consistencia**) pero no se permite una pérdida de comunicación entre los nodos (**Tolerancia a particiones**).

3.2.5 Estado del arte de la investigación

En este estudio de investigación Ciges, J.M. (2012) que probablemente pueda ser uno de los pocos artículos que se ha enfocado a estudiar a fondo el comportamiento transaccional y funcional entre un motor de base de datos relacional como lo es MySQL y una base de datos tipo documental MongoDB, se establece un marco de evaluación donde se analizó los log de acceso a internet generados por una gran cantidad de usuarios, el objetivo de este análisis estuvo encaminado a responder una serie de preguntas tales como: ¿ Cuáles son las páginas más visitadas?, ¿ Qué usuarios pasan más tiempo en línea?, ¿ Cuáles son los 100 usuarios cuyo volumen de tráfico es mayor?, ¿Cuál es el volumen promedio diario de tráfico de la red corporativa a través de internet?. El comparativo permitió identificar el mayor desempeño entre los dos motores de bases de datos y se convierte en una base fundamental para el desarrollo de este proyecto.

El proceso consistió inicialmente en ingresar la mayor cantidad de datos en cada uno de los motores divididos en la siguiente información: 70.000 usuarios visitantes, 70.000 direcciones IP de visitantes, 1.300.000 dominios de internet, 90.000.000 de entradas de registro no FTP y 4.500.000 entradas de registro FTP, esta información se ingresó de manera aleatoria y se realizó a través de 3 clases PHP que se convirtió en el lenguaje de programación elegido para llevar a cabo este desarrollo. Los test de comparación se agruparon en:

- Test de inserción
- Test de lectura simultanea multiusuario
- Test de escritura simultanea multiusuario
- Test de lectura de consultas complejas (Agregación)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Los resultados fueron los siguientes:

Tabla 1: Test simples de lectura

	MySQL	MongoDB
Leer todos los objetos con id entre 100 y 110 millones	10m 5s	0,15s
Seleccionar un registro por id (entre 113 millones)	0,30s	0,12s
Seleccionar los ids de los registros entre dos fechas	4m 10s	0,16s
Seleccionar los ids de los registros entre dos fechas y contarlos	2m 48s	0,85s

Tabla 2: Test simples de escritura

	MySQL	MongoDB
70.000 usuarios	12s	3s
1.300.000 dominios	4m 36s	58s
1.300.000 dominios con índices	14m 13s	8m 27s
5.000.000 de entradas en el log	2h 10m 54s	1h 03m 53s
10.000.000 de entradas en el log	3h 27m 10s	1h 59m 11s
30.000.000 de entradas en el log	10h 18m 46s	5h 55m 25s

Tabla 3: Test de lectura de consultas complejas (Agregación)

Consultas	MySQL	MongoDB
10 dominios con los totales de visita más altos	13m 13s	2m 37s
10 usuarios con más accesos a Internet	24m 02s	3m 53s
Tráfico medio Internet para junio	12m 05s	2m 42s

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Después de analizados los resultados de los test anteriores el estudio concluye que MongoDB es más rápido en las operaciones de escritura, cabe anotar que para operaciones de escritura sencillas es entre 2 y 4 veces más rápido y para un número elevado de escrituras aproximadamente 30 millones por ejemplo MongoDB duplica en rendimiento a MySQL. En los test de lectura se puede apreciar que MongoDB continua siendo más rápido. Por último los test de agregación donde se puso a prueba el marco nativo de agregación de MongoDB sobre el de MySQL demostraron que este último supera ampliamente a MongoDB entre 3 y 6 veces más rápido.

En este artículo Sobreira, G. (2010) nos encontramos con un análisis comparativo teórico entre el teorema CAP con sus bases de datos NoSQL y las bases de datos relacionales, en el cual se puede evidenciar una vez más los factores limitantes de este último frente a un fenómeno cada vez más presente en las organizaciones y en la sociedad interconectada en general donde la estructuración de los datos, la poca flexibilidad y el continuo crecimiento de los mismos se convierten en el desafío principal.

El trabajo de investigación plantea un análisis comparativo entre una base de datos relacional PostgreSQL y una base de datos NoSQL CouchDB, donde se demuestra que el Teorema CAP (consistencia, disponibilidad y tolerancia a partición) representa una verdadera alternativa aplicada a la construcción de entornos distribuidos donde los Sistemas de Gestión de Base de Datos (DBMS) se convierten en una tarea compleja y difícil de alcanzar.

El desarrollo del trabajo se distribuye en 7 capítulos, en los cuales se logra abarcar todos los conceptos y comparaciones teóricas que dan respuesta al objetivo propuesto en esta investigación. Se inicia con los conceptos introductorios de un sistema distribuido, sus ventajas y desventajas, continuando con la explicación del teorema CAP el cual es el eje central del análisis, para luego en los dos capítulos siguientes abarcar los conceptos del modelo relacional y las principales características de la base de datos relacional PostgreSQL.

En un quinto capítulo se discute las bases fundamentales del paradigma NoSQL, haciendo un análisis comparativo con el modelo relacional, para luego dar paso al capítulo donde se explica la elección de CouchDB como representante de las bases de datos NoSQL al igual de aspectos tales como su modelo de datos y la forma de recuperación entre otras características

Por último se concluye con el objeto de la investigación, el análisis comparativo del teorema CAP entre la base de datos relacional PostgreSQL y la base de datos NoSQL CouchDB.

El estudio concluye que mientras que PostgreSQL tiene como objetivo proporcionar la consistencia y la disponibilidad absoluta, CouchDB da prioridad a la disponibilidad y propiedades de partición tolerante, renunciando a la coherencia por lo tanto permite mejoras de rendimiento más altos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Llevando los resultados de la investigación a un entorno aplicado como es la situación de tomar la decisión de qué solución sería la más adecuada a implementar en una empresa, se considera importante identificar la necesidad de la aplicación y no descartar la posibilidad de utilizar las dos soluciones en conjunto para satisfacer los objetivos establecidos.

En Brito Zhunio, D.M, (2011), tesis Orientada al estudio de NoSQL como alternativa y complemento de un gestor de bases de datos específicamente en aplicaciones web, se realiza un trabajo de investigación donde se describe uno a uno los pasos necesarios para poner en funcionamiento de forma general el motor de bases de datos MongoDB. Si bien el estudio explica detalladamente las fases de configuración y diseño de una colección de datos en MongoDB, no se realiza un análisis técnico que determine si la tecnología NoSQL satisface los requerimientos de las aplicaciones de software orientadas a entornos web.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

METODOLOGÍA PROPUESTA

El desarrollo y puesta en marcha del comparativo de rendimiento entre los dos motores de bases de datos mencionados anteriormente, está descrito paso a paso en las siguientes fases de implementación.

Tabla 4: Cronograma

FASE	OBJETIVO	ACTIVIDADES
FASE 1. Caracterización	Caracterizar los modelos Relacional y NoSQL y los precedentes en cuanto su comparación.	1.1 Realizar una revisión bibliográfica acerca del modelo relacional caracterizando sus propiedades y limitaciones. 1.2 Realizar una revisión bibliográfica sobre el modelo NoSQL definiendo sus características y limitaciones. 1.3 Hacer una búsqueda bibliográfica de los trabajos relacionados que comparan los modelos de bases de datos Relacional y NoSQL y encontrar sus limitaciones.
FASE 2 Diseño de pruebas	Establecer métricas y pruebas que permitan identificar el rendimiento entre los modelos Relacional y NoSQL.	2.1 Identificar y definir las métricas mediante las cuales se realizará la comparación entre ambos modelos. 2.2 Diseñar las pruebas por medio de las cuales se validan el modelo relacional y NoSQL con las métricas definidas.
FASE 3 Diseño y Desarrollo del caso de estudio	Diseñar y desarrollar un software para la gestión de información de contacto de un grupo de personas que permita validar el modelo relacional y NoSQL basado en las métricas establecidas.	3.1 Diseño de un sistema de software para el almacenamiento de la información de contacto de un grupo de personas 3.2 Desarrollo e implementación del sistema propuesto para la gestión de personas
FASE 4 Evaluación	Evaluar los resultados de la comparación entre el modelo relacional y NoSQL en proceso de almacenamiento y consulta de un	4.1 Recopilación de los datos arrojados por las pruebas aplicadas en el sistema para la gestión de personas. 4.2 Evaluación y análisis de resultados de la comparación entre los modelos

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

	grupo de personas	Relacional y NoSQL.
--	-------------------	---------------------

Pruebas de rendimiento

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación. En la fase de diseño de pruebas se deben definir e identificar las métricas necesarias mediante las cuales se realizará la comparación entre los dos modelos.

Los parámetros de medición que permitirán identificar el nivel de rendimiento entre los dos motores de bases de datos estarán divididos en dos grupos: uno que tienen relación con el sistema operativo y sus componentes y otro que es propio del motor de bases de datos el cual arrojará los tiempos de ejecución entre cada sentencia de inserción y consulta sobre la base de datos.

5.1 Métricas de Sistema Operativo:

Estas métricas están orientadas a identificar el tráfico de red, la entrada y salida (E/S) en disco y el tiempo de CPU que se genera en cada consulta, inserción, eliminación y actualización en ambos motores de bases de datos. A través del monitor de rendimiento de Windows (Perfmon.exe) se monitorea el consumo de recursos del sistema en el momento que se lancen las diferentes transacciones, al terminar la ejecución del conjunto recopilador de datos configurado previamente se obtiene un informe donde se detalla los tiempos de respuesta del sistema. Los siguientes son los indicadores que permitirán identificar a nivel de sistema operativo que motor tiene un mayor rendimiento.

- **Entrada/Salida (E/S) De Disco:**

Los discos tienen una velocidad de entrada/salida. Cuando se requieren datos a una velocidad mayor que ésta, el disco se convierte en un cuello de botella. A nivel de operaciones de entrada/salida sobre el disco tenemos los siguientes identificadores.

Lecturas de disco/seg: Comprende las operaciones de lectura sobre el disco duro en el momento ejecutar una instrucción sobre los motores de bases de bases de datos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Escritura de disco/seg: Comprende las operaciones de escritura sobre el disco duro en el momento ejecutar una instrucción sobre los motores de bases de bases de datos.

- **Ciclos de CPU:** Un uso excesivo continuo de la CPU puede indicar que las consultas de las diferentes transacciones pueden no estar optimizadas. Los indicadores asociados al rendimiento de la CPU son los siguientes:

% de tiempo de uso del procesador: Describe la supervisión de una o varias instancias efectuadas sobre el motor de base de datos para verificar si los índices de CPU son normales

- **Uso de Memoria Ram:** El objetivo de este medidor es identificar a través de las transacciones que se generan el uso de la memoria RAM asignada y visualizar si existen fallos de paginación. Para nuestro análisis se tendrán en cuenta los siguientes indicadores:

Memoria: Bytes disponibles: Indica la memoria actual disponible para los procesos en ejecución.

Memoria: Páginas/seg: Permite identificar si existe algún tipo de presión sobre la memoria a causa de la memoria usada por el motor de bases de datos

5.2 Métricas de Base datos

Por otro lado estas métricas hacen parte del conjunto de pruebas que buscan identificar los tiempos entre cada operación de inserción, eliminación y actualización sobre los motores de bases de datos, y que se obtienen a través del aplicativo desarrollado para la gestión de transacciones sobre las bases de datos, se elegirán los siguientes indicadores para cumplir este objetivo:

- **Tiempo de ejecución en una operación de inserción sobre la base de datos**
- **Tiempo de ejecución en una operación de eliminación sobre la base de datos**
- **Tiempo de ejecución en una operación de actualización sobre la base de datos**
- **Consultas con creación de índices donde se determine los tiempos de ejecución cuando la base de datos se bloquea por periodos de tiempo más extensos**
- **Tiempos de ejecución sobre funciones de agregado que nos permita determinar el rendimiento sobre un conjunto de valores.**

Desarrollo e implementación del aplicativo

La fase 3 del proyecto establece el desarrollo de un sistema de software a través del cual se validara el modelo relacional y NoSQL basado con las métricas establecidas. A continuación se describe los modelos, diagramas y funcionalidades del aplicativo.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PROCESO DE DESARROLLO DE SOFTWARE

The Institute of Electrical and Electronics Engineers (IEEE) define ingeniera de software como la aplicación sistemática, disciplinada y cuantificable de un proceso de desarrollo, operación y mantenimiento de un producto software.

El prototipo de software permite manipular las acciones de inserción, actualización, eliminación, búsqueda y consulta de datos sobre una alta cantidad aleatoria de personas.

Dado su alto desempeño, portabilidad y excelente representante del paradigma de programación orientado a objeto, el aplicativo fue desarrollado bajo el lenguaje Java, donde se establecieron las respectivas clases, funciones y procedimientos necesarios para la generación de datos, materia prima de este proyecto de investigación.

Como entorno de desarrollo se utilizó el software Netbeans en su versión 7.3.1, con éste se simplificó la gestión del proyecto de desarrollo ya que ofrece características como el ser multilenguaje, con un amplio rango de tecnologías de desarrollo y fundamentalmente por su soporte para los dos motores de bases de datos objetos de comparación.

Análisis y Diseño

El objetivo del aplicativo es gestionar la información de contacto de una cantidad aleatoria de personas, tanto para motor de base de datos MySQL como para el de MongoDB. En la figura 4 se describe la entidad Persona con sus respectivos atributos.

PERSONA
-id
-Cedula
-Nombres
-Apellidos
-CiudadExpID
-CiudadNto
-FechaNto
-RH
-GrpSanguineo
-Sexo
-Ciudad
-Direccion

Figura 4: Entidad Persona

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Implementación

El aplicativo está distribuido en un conjunto de paquetes, los cuales tienen un propósito específico en el desarrollo del software, la figura 5 describe las principales características de cada uno de estos.

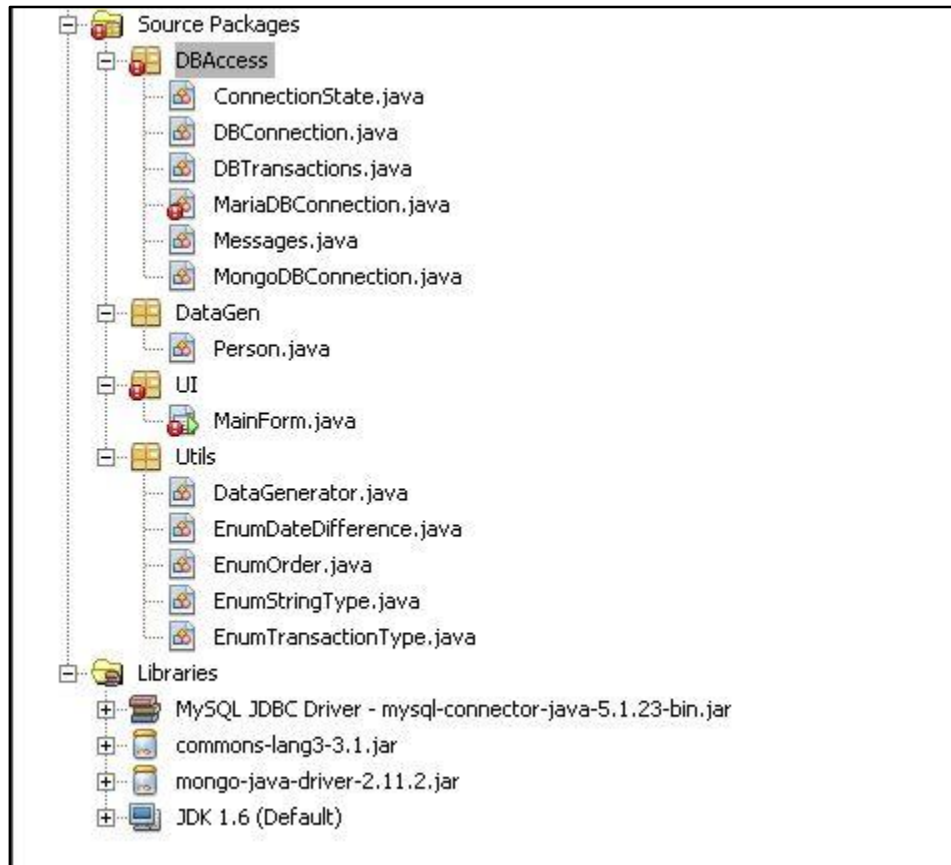


Figura 5: Paquetes de fuente

- **DBAccess:** Este paquete contiene las clases que corresponden al acceso a los motores de bases de datos tanto de MySQL como MongoDB, al igual que sus atributos, propiedades y métodos públicos como privados que permiten la ejecución de las diferentes transacciones de Insert, Update, Delete, Aggregate y Search.
- **DataGen:** En este Paquete se encuentra la clase *Person*, en la cual están definidos los atributos propios y los métodos tanto de obtención (*get*), como de asignación (*set*) de los diferentes **objetos persona** que se gestionan en el aplicativo.
- **UI:** El formulario principal y la interfaz gráfica de usuario se encuentran en el paquete **UI**, desde esta clase se gestionan los diferentes eventos de cada componente que el usuario acciona de acuerdo a la transacción que requiera ejecutar.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- **Utils:** En el paquete **Utils** se almacenan aquellas clases complemento del desarrollo que facilitan entre otras la generación aleatoria de registros, las capturas de tiempos de respuesta y la definición del tipo de transacciones a utilizar.
- **Libraries:** Las librerías están contenidas en este último paquete para permitirnos identificar, definir y gestionar las diferentes clases, métodos y atributos propios del motor de bases de datos en nuestro lenguaje de programación java.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSIÓN

RESULTADOS

Materiales y condiciones con que se desarrollaron las pruebas de rendimiento

Para realizar las pruebas de rendimiento entre los dos motores de bases de datos se utilizó un equipo con las siguientes características descritas en la siguiente tabla:

Característica	Descripción
Procesador	Intel core i3 Segunda generación
Memoria RAM	4 GB
Sistema Operativo	Windows 7 Profesional de 64 bits

Tabla 5: Características del Equipo

Se instaló una plataforma de software libre APPSERV, la cual contiene un servidor web Apache, una base de datos MySQL y un intérprete para el lenguaje (PHP), las versiones de este software se describen a continuación:

Software	Versión
APPSERV	2.6.0
APACHE	2.2.8
PHP	6.0.0
MYSQL	6.0.4

Tabla 6: Características servidor web

Para la gestión de datos en la plataforma NoSQL se instaló MongoDB en su versión 3.2.1

Consideraciones adicionales:

Para el seguimiento de las transacciones ejecutadas sobre el motor de base de datos MySQL se utilizó la herramienta PHPMYADMIN en su versión 2.10.3, desde la cual se verifica que los datos afectados sobre la tabla correspondiente estuvieran dentro de los parámetros definidos, de igual forma para el monitoreo de transacciones sobre el motor de base de datos NoSQL se utilizó la línea de comandos de Windows (command), donde a través de líneas de comandos propias de la plataforma MongoDB se hace el seguimiento respectivo.

Resultados de las pruebas de rendimiento

A continuación se muestran de forma detallada cada uno de los resultados obtenidos de las

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

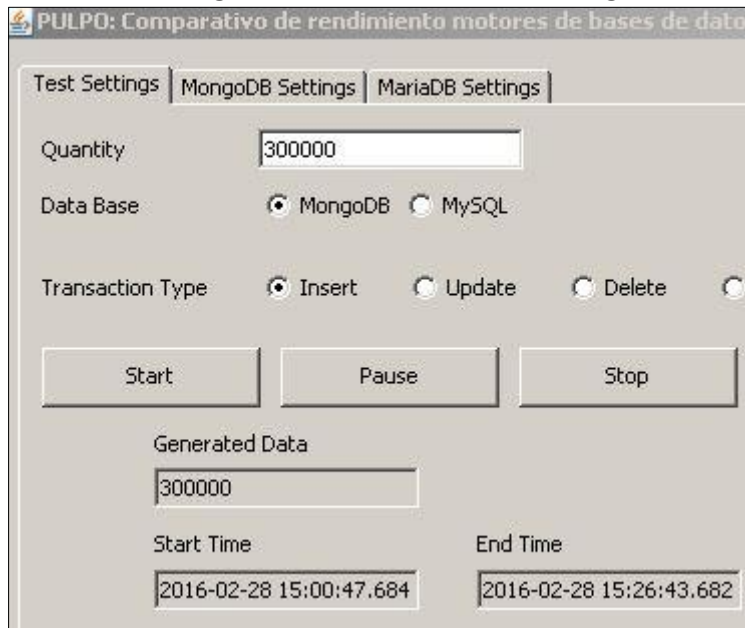
respectivas pruebas de rendimiento definidas en el alcance de este proyecto.

Pruebas de inserción

De un total de 300.000 registros aleatorios generados por el aplicativo, se obtuvieron los siguientes tiempos de inserción:

- **MySQL:** 1h:27min:10seg (1 hora, 27 minutos, 10 segundos)
- **MongoDB:** 27min:36seg (27 minutos, 36 segundos)

Figura 6: Resultado inserción MongoDB



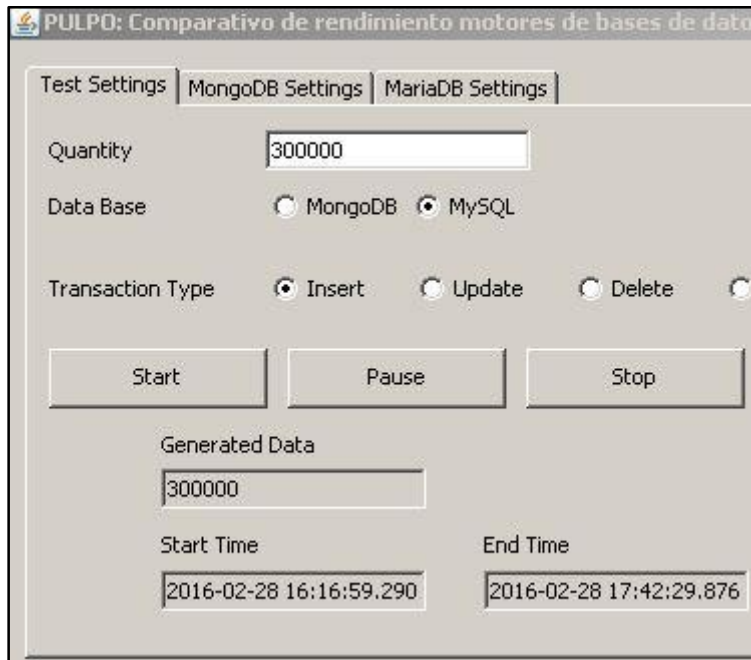


Figura 7: Resultado inserción MySQL

Pruebas de Actualización

Para efectos del comparativo se actualizó los campos sexo y ciudad de un total de 300 mil registros almacenados en ambos motores de bases de datos, se realizaron dos test obteniendo los mismos resultados, para tal transacción se obtuvieron los siguientes tiempos de ejecución.

1 test

- MySQL: 13seg (13 segundos)
- MongoDB: 12 seg (12 segundos)

Figura 8: Resultado actualización MongoDB test 1

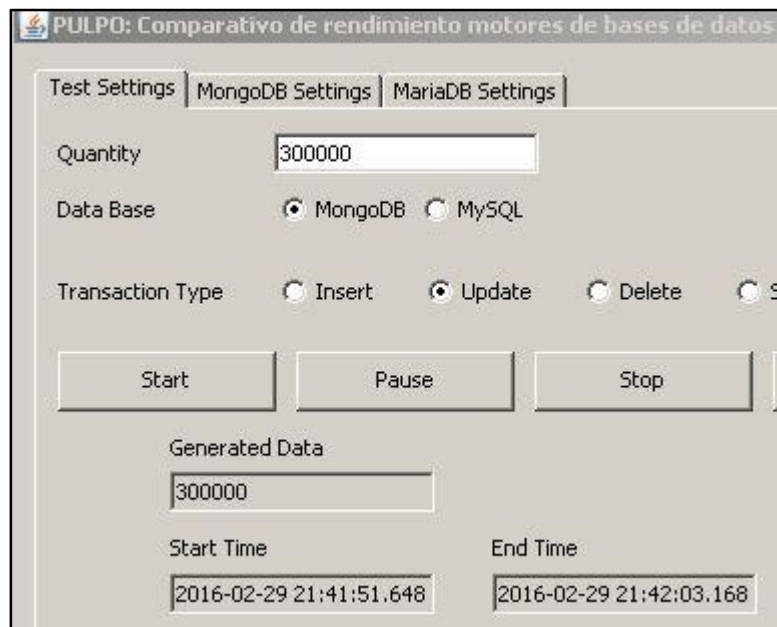
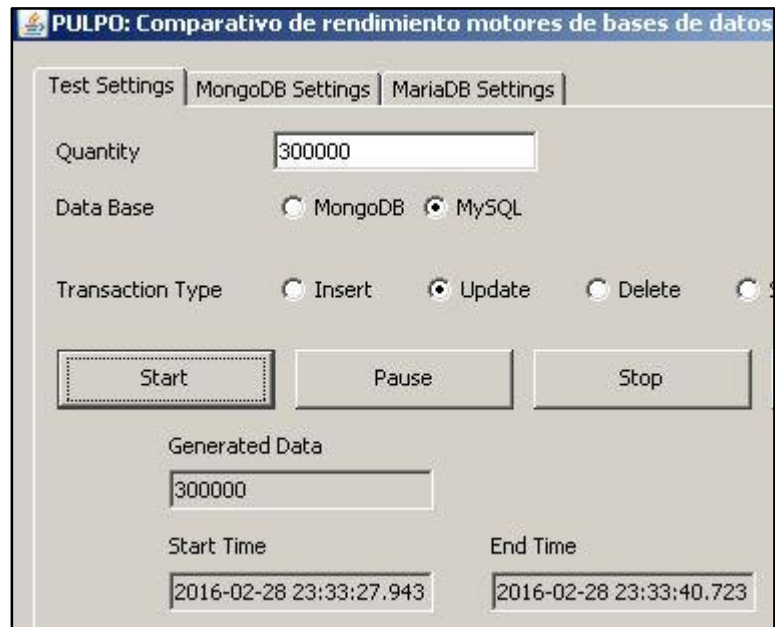


Figura 9: Resultado actualización MySQL test 1



2 test

- MySQL: 13seg (13 segundos)
- MongoDB: 10 seg (10 segundos)

Figura 10: Resultado actualización MongoDB test 2

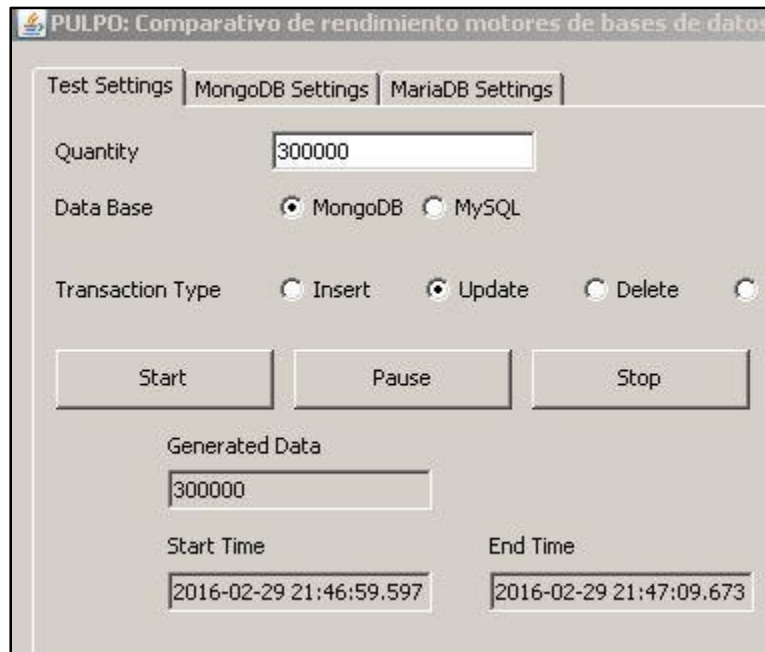
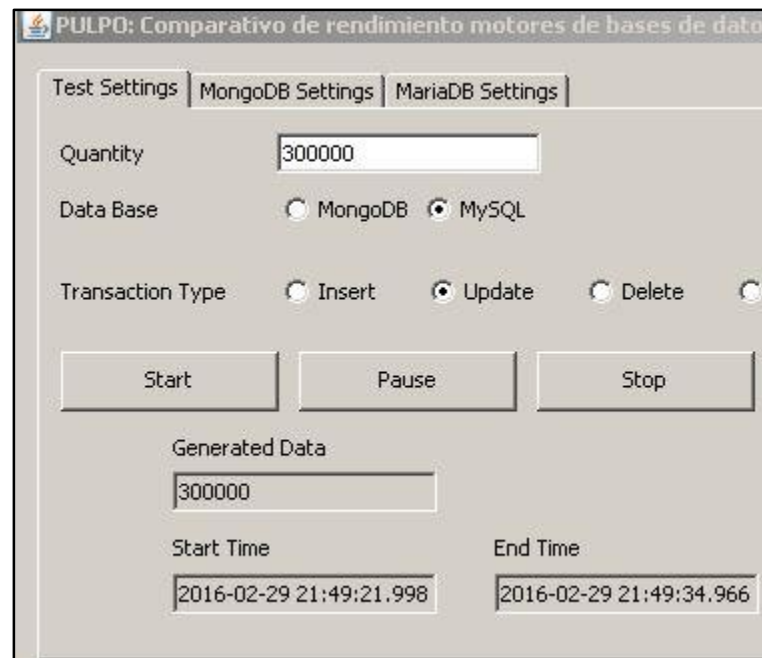


Figura 11: Resultado actualización MySQL test 2



Pruebas de búsqueda

La transacción de búsqueda consiste en ordenar 300 mil registros en forma ascendente, al igual que la anterior prueba se realizaron dos test para la transacción de búsqueda, los resultados se muestran a continuación:

1 test

- MySQL: 2 seg (2 segundos)
- MongoDB: 1 seg (1 segundo)

Figura 12: Resultado búsqueda MongoDB test 1

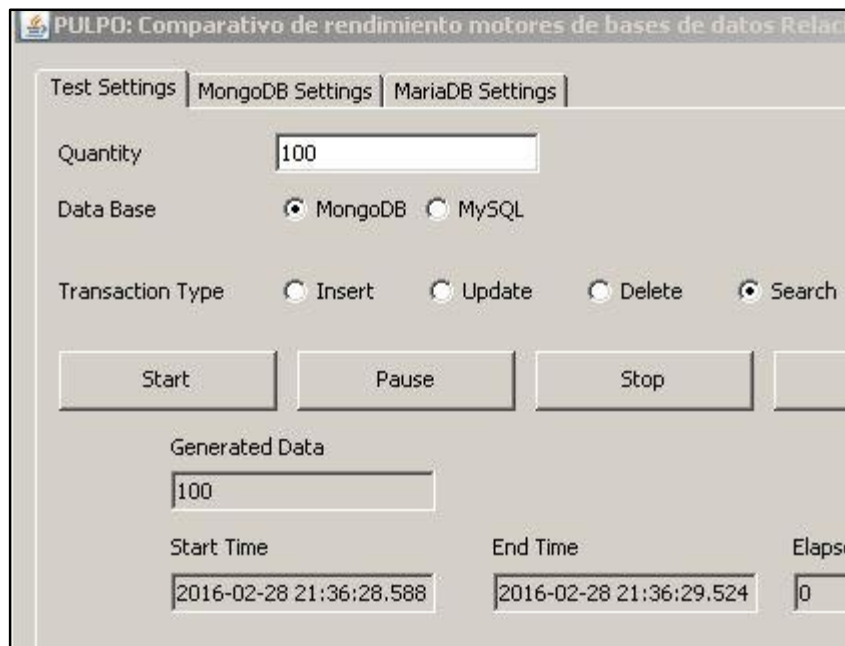
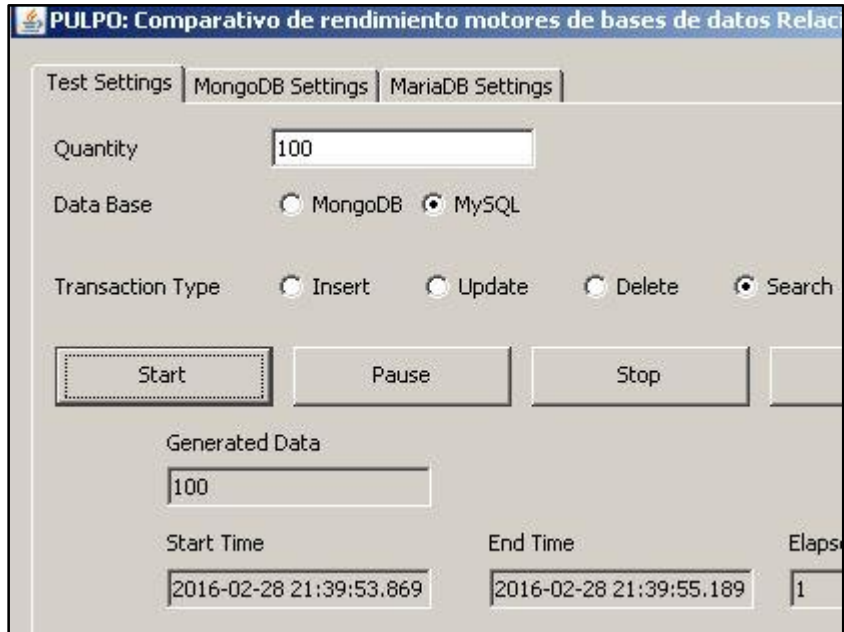


Figura 13: Resultado búsqueda MySQL test 1



2 test

- MySQL: 2 seg (2 segundos)
- MongoDB: 18 milisegundos (0.018 segundos)

Figura 14: Resultado búsqueda MongoDB test 2

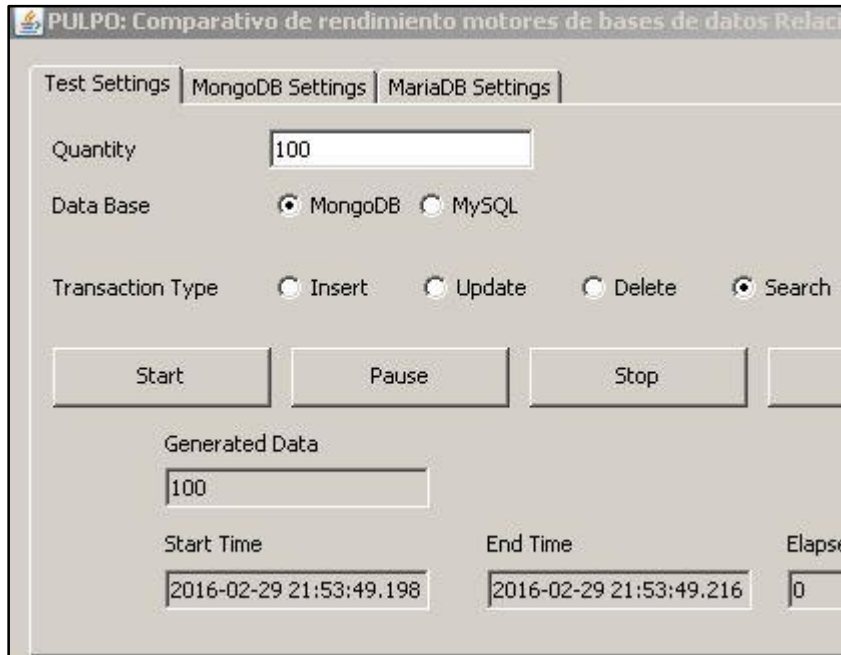
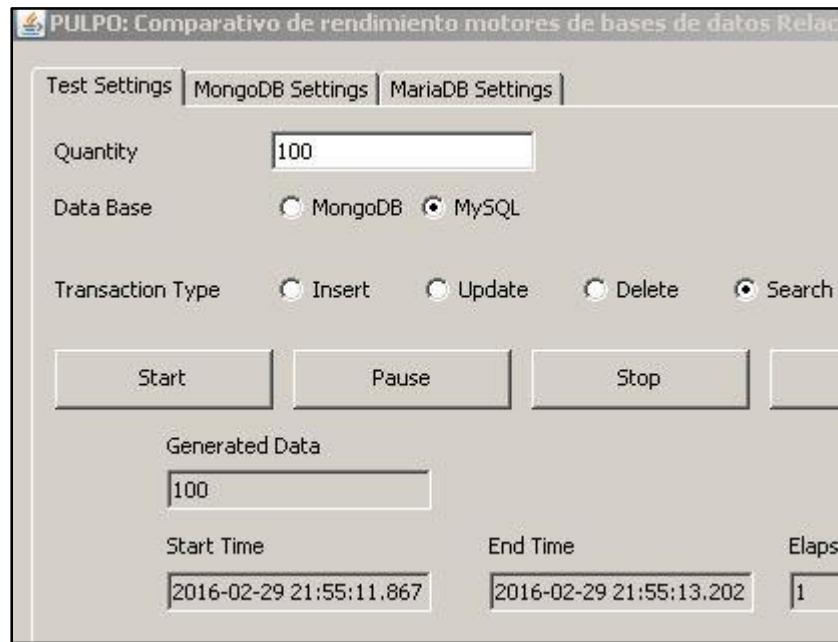


Figura 15: Resultado búsqueda MySQL test 2



Pruebas de funciones de agregado

Para estas pruebas de rendimiento, las funciones de agregado son importantes en la medida que permiten identificar la respuesta de los motores de bases de datos frente a un cálculo sobre un conjunto de 300 mil registros, a continuación veremos los resultados obtenidos:

1 test

- MySQL: 2 seg (2 segundos)
- MongoDB: 6 seg (6 segundos)

Figura 16: Resultado función de agregado MongoDB test 1

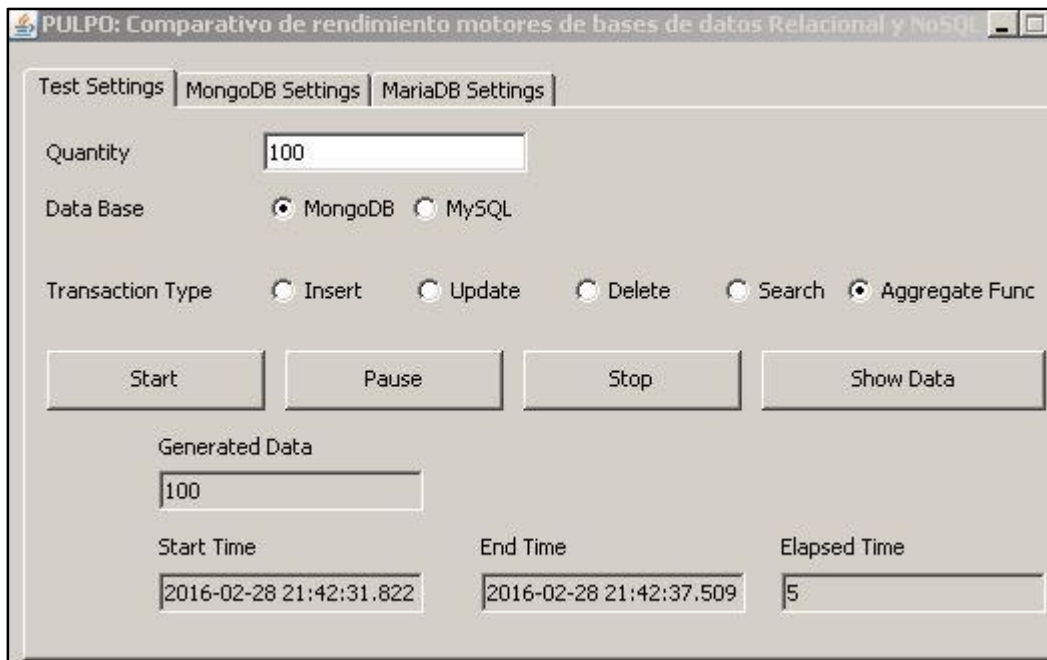
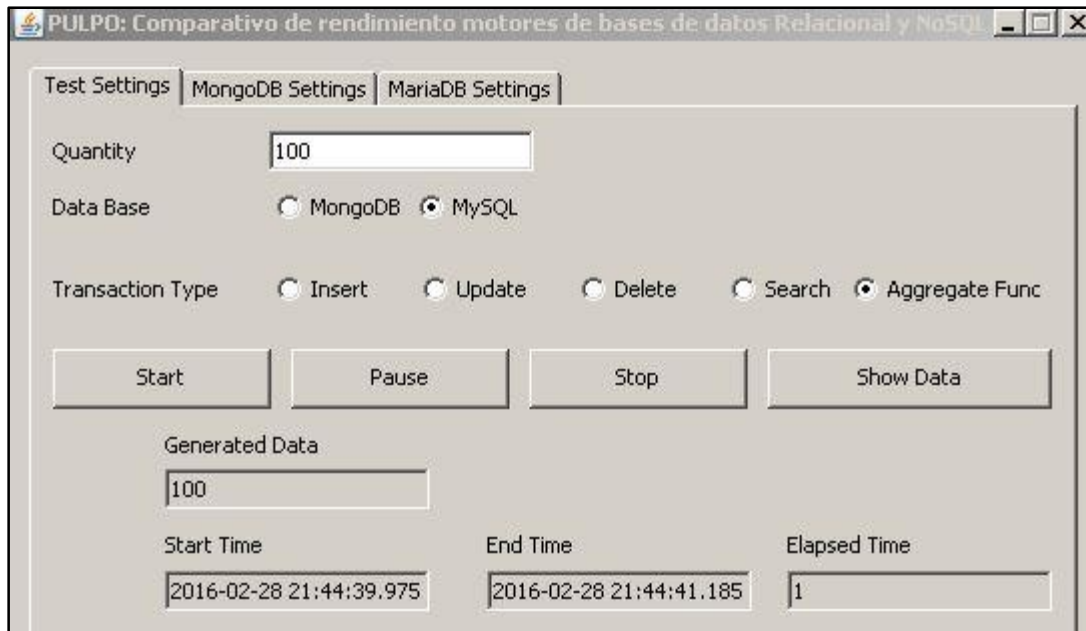


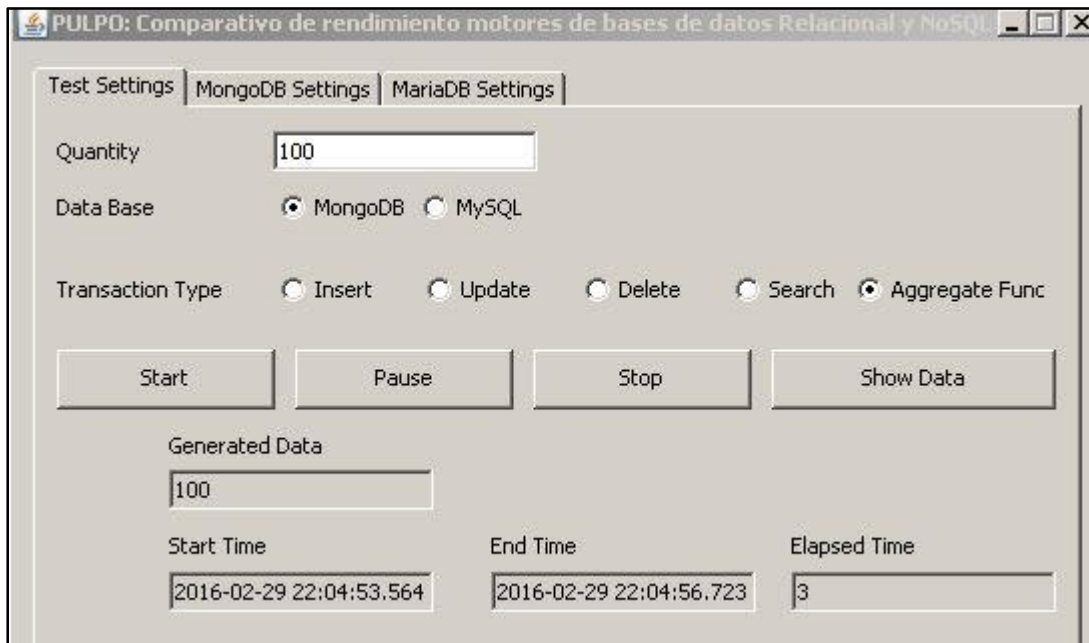
Figura 17: Resultado función de agregado MySQL test 1



2 test

- MySQL: 2 seg (2 segundos)
- MongoDB: 3 seg (3 segundos)

Figura 18: Resultado función de agregado MongoDB test 2



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Figura 19: Resultado función de agregado MySQL test 2



Pruebas de Eliminación

Por último tenemos los resultados sobre las pruebas de eliminación de una cantidad de registros de 300 mil.

1 test

- MySQL: 11 seg (11 segundos)
- MongoDB: 9 seg (9 segundos)

Figura 20: Resultado de eliminación MongoDB

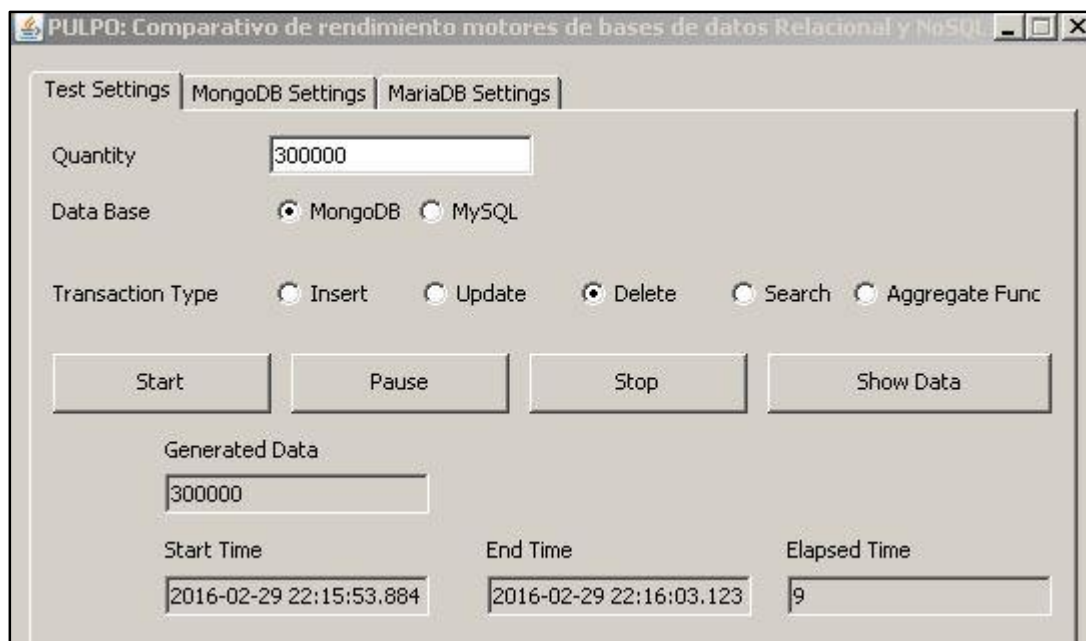
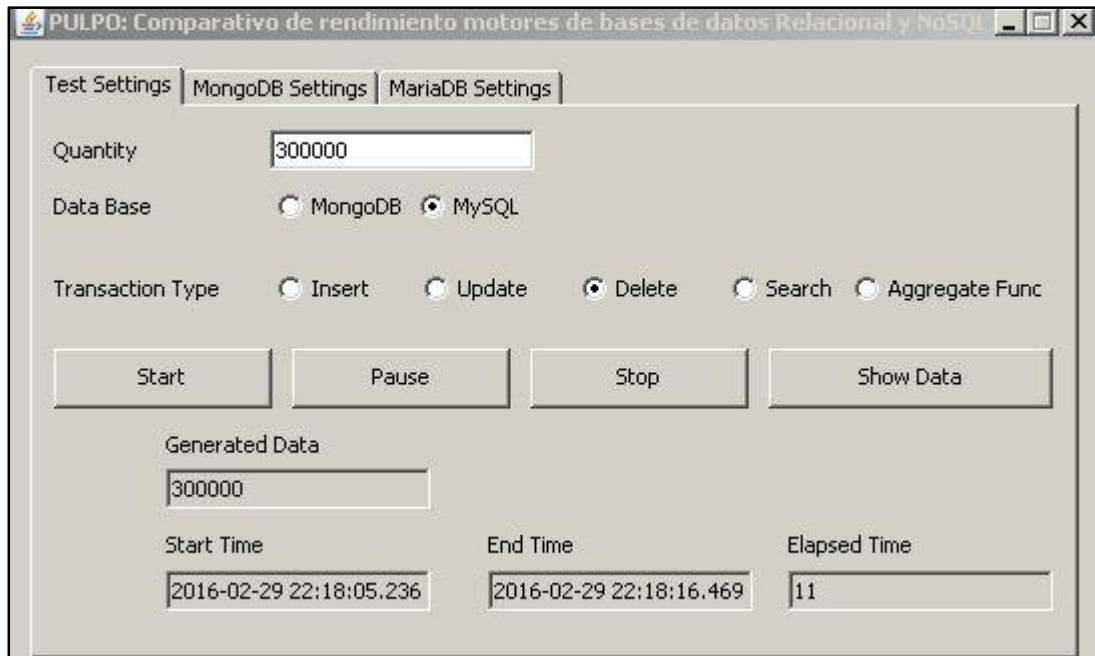


Figura 21: Resultado de eliminación MySQL



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Rendimiento de sistema operativo

A continuación haremos un seguimiento al rendimiento del sistema operativo en las pruebas de inserción sobre los motores de bases de datos MongoDB y MySQL, pasaremos de un estado en reposo del sistema a un estado donde se esforzará sus componentes en la inserción de registros en MongoDB, luego capturamos nuevamente los datos del sistema en su estado normal para después pasar a registrar los datos de rendimiento del sistema bajo la inserción de registros en MySQL.

En condiciones normales tal como observamos en las figuras 22 y 23 el equipo utilizado para las pruebas tuvo los siguientes rendimientos en su sistema operativo antes de realizar la inserción de 300 mil registros.

Resumen		Disco	
Proceso		Disco principal por velocidad de E/S:	0
Total de % de CPU:	9	ES/s:	6
Grupo de proceso superior:	System	Longitud de la cola de disco:	3
% de CPU de grupo:	4		
Total de % de CPU:	1		
Grupo de proceso superior:			

Figura 22: indicadores del sistema en reposo

Memoria		Red	
Uso:	52 %	Uso:	0 %
Memoria:	2934 MB	Cliente de salida más usado:	127.0.0.1
Proceso más frecuente:	svchost##3	Enviado:	1,023 bytes
Espacio de trabajo privado:	140,300 KB	Cliente de entrada más usado:	178.62.93.181
		Recibido:	145 bytes

Figura 23: indicadores del sistema complementarios en reposo

Iniciadas las pruebas de inserción sobre la base de datos de MongoDB el sistema operativo tuvo los siguientes rendimientos registrados en las figuras 24 y 25.

Resumen		Disco	
Total de % de CPU:	32	Disco principal por velocidad de E/S:	0
Grupo de proceso superior:	java.exe	ES/s:	23
% de CPU de grupo:	12	Longitud de la cola de disco:	4
Total de % de CPU:	1		
Grupo de proceso superior:			

Figura 24: indicadores del sistema en prueba de inserción MongoDB

Memoria		Red	
Uso:	65 %	Uso:	0 %
Memoria:	2934 MB	Cliente de salida más usado:	127.0.0.1
Proceso más frecuente:	java	Enviado:	70,363 bytes
Espacio de trabajo privado:	413,052 KB	Cliente de entrada más usado:	127.0.0.1
		Recibido:	94,549 bytes

Figura 25: indicadores del sistema complementarios en prueba de inserción MongoDB

Nuevamente en reposo el sistema arroja los siguientes rendimientos en sus componentes que podemos verificar en las figuras 26 y 27.

Resumen		Disco	
Total de % de CPU:	8	Disco principal por velocidad de E/S:	0
Grupo de proceso superior:	System	ES/s:	10
% de CPU de grupo:	4	Longitud de la cola de disco:	3
Total de % de CPU:	1		
Grupo de proceso superior:			

Figura 26: indicadores del sistema luego de prueba de inserción MongoDB

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Memoria		Red	
Uso:	33 %	Uso:	0 %
Memoria:	2934 MB	Cliente de salida más usado:	127.0.0.1
Proceso más frecuente:	svchost##3	Enviado:	1,018 bytes
Espacio de trabajo privado:	137,828 KB	Cliente de entrada más usado:	127.0.0.1
		Recibido:	116 bytes

Figura 27: indicadores del sistema complementario luego de prueba de inserción MongoDB

En las figuras 28 y 29 podemos observar los rendimientos que presentó el sistema operativo para la prueba de inserción sobre el motor de base de datos MySQL.

Resumen		Disco	
Total de % de CPU:	27	Disco principal por velocidad de E/S:	0
Grupo de proceso superior:	mysqld.exe	ES/s:	126
% de CPU de grupo:	8	Longitud de la cola de disco:	76
Total de % de CPU:	1		
Grupo de proceso superior:			

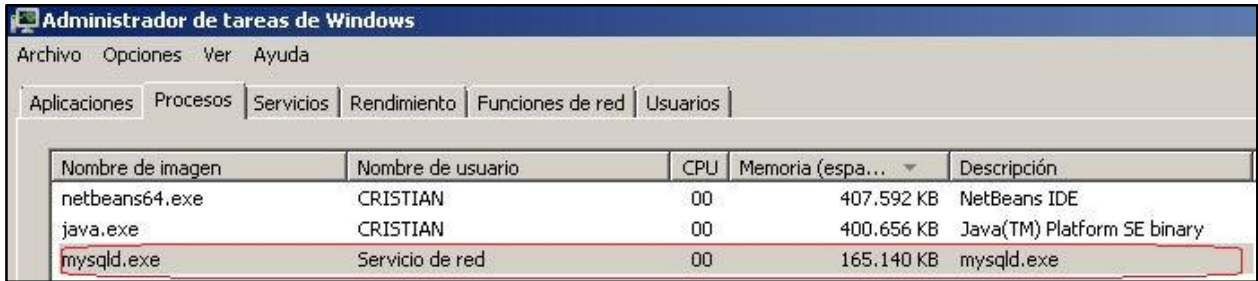
Figura 28: indicadores del sistema en prueba de inserción MySQL

Memoria		Red	
Uso:	73 %	Uso:	0 %
Memoria:	2934 MB	Cliente de salida más usado:	127.0.0.1
Proceso más frecuente:	netbeans64	Enviado:	456,731 bytes
Espacio de trabajo privado:	419,348 KB	Cliente de entrada más usado:	127.0.0.1
		Recibido:	81,639 bytes

Figura 29: indicadores del sistema complementario en prueba de inserción MySQL

En la figura 30 podemos observar el consumo de memoria RAM del proceso mysqld.exe durante las pruebas de inserción.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Administrador de tareas de Windows

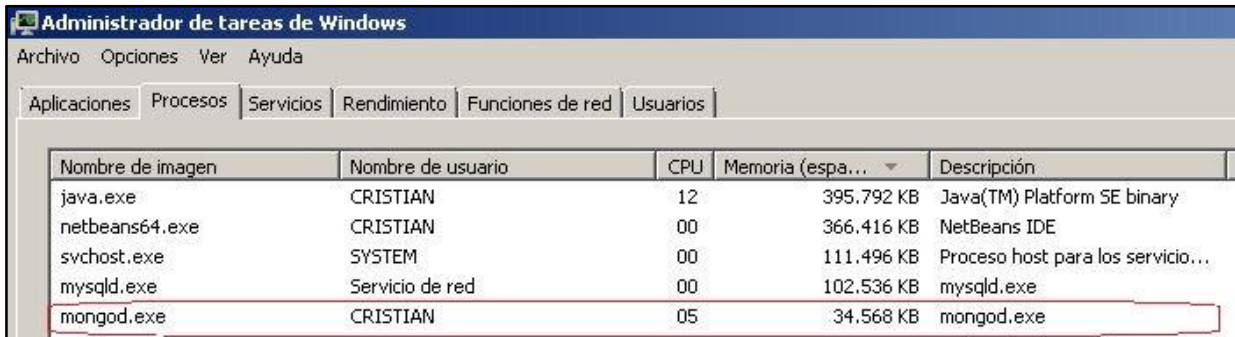
Archivo Opciones Ver Ayuda

Aplicaciones Procesos Servicios Rendimiento Funciones de red Usuarios

Nombre de imagen	Nombre de usuario	CPU	Memoria (espa...)	Descripción
netbeans64.exe	CRISTIAN	00	407.592 KB	NetBeans IDE
java.exe	CRISTIAN	00	400.656 KB	Java(TM) Platform SE binary
mysqld.exe	Servicio de red	00	165.140 KB	mysqld.exe

Figura 30: Consumo de memoria RAM proceso Mysqld.exe

El proceso mongod.exe al ejecutarse las mismas pruebas de inserción muestra el siguiente consumo de memoria RAM plasmado en la figura 31.



Administrador de tareas de Windows

Archivo Opciones Ver Ayuda

Aplicaciones Procesos Servicios Rendimiento Funciones de red Usuarios

Nombre de imagen	Nombre de usuario	CPU	Memoria (espa...)	Descripción
java.exe	CRISTIAN	12	395.792 KB	Java(TM) Platform SE binary
netbeans64.exe	CRISTIAN	00	366.416 KB	NetBeans IDE
svchost.exe	SYSTEM	00	111.496 KB	Proceso host para los servicio...
mysqld.exe	Servicio de red	00	102.536 KB	mysqld.exe
mongod.exe	CRISTIAN	05	34.568 KB	mongod.exe

Figura 31: Consumo de memoria RAM proceso Mongod.exe

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

El desarrollo de este trabajo de investigación nos muestra dos tecnologías de bases de datos, por un lado el modelo Relacional con sus características ACID conservando la fiabilidad en sus transacciones y por el otro el paradigma NoSQL que pretende ser una alternativa de gestión de datos donde el modelo tradicional tiene sus limitantes. Sometidos ambos modelos a un ambiente de pruebas de rendimiento bajo el mismo lenguaje de programación, entorno de desarrollo y lógica de programación nos encontramos en general con que el motor de bases de datos MongoDB es mejor opción que el motor de base de datos MySQL en aquellos entornos de desarrollo donde se necesite intensidad y velocidad en la lectura y escritura de datos como los entornos web actuales, sin embargo si lo que se requiere es un soporte transaccional integral conservando las características ACID es recomendable seguir usando un sistema RDBMS. Las tablas 7 y 8 nos muestran los rendimientos consolidados tanto en las métricas de bases de datos como en las de sistema operativo para ambos motores.

Tabla 7: Resultado pruebas de rendimiento, Métricas de Bases de Datos

Operación (Tiempo)	Inserción	Actualización	Búsqueda	Función de agregado	Eliminación
MySQL	1 hora, 27 minutos, 10 segundos	13 segundos	2 segundos	2 segundos	11 segundos
MongoDB	27 minutos, 36 segundos	10 segundos	0.018 segundos	6 segundos	9 segundos

Tabla 8: Resultado pruebas de rendimiento, Métricas de sistema operativo

Rendimiento sistema operativo en pruebas de Inserción				
Métrica	Uso CPU (%)	Uso Memoria RAM (%)	Actividad de Disco (ES/s)	Consumo memoria RAM servicio (MB)
MySQL	27 %	73 %	126 MB/s	Mysqld.exe (165 MB)
MongoDB	32 %	65 %	23 MB/s	Mongod.exe (34 MB)

Si bien existen aplicaciones web tales como Facebook, Twitter entre otras que dentro de sus arquitecturas utilizan motores de bases de datos NoSQL como Cassandra, MongoDB, para sus publicaciones y contenidos, en la actualidad no se tiene una amplia documentación y peor aún conocimiento frente a lo que es el paradigma NoSQL dificultando de esta forma recopilar antecedentes relevantes que soporten comparativos previos de rendimiento.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Es importante resaltar que en la actualidad exista un lenguaje de programación como lo es java que desde su portabilidad y robustez nos permita combinar estos dos motores de bases de datos en una sola plataforma de desarrollo, lo cual facilita para futuros investigadores que se interesen en este tema el llevar a cabo avances y mejoras que en este trabajo no se hayan tenido en cuenta.

Los rendimientos del equipo y la gestión de recursos del sistema operativo donde se realizaron las pruebas nos muestran una diferencia sin ser muy amplia en cuanto al consumo de memoria RAM, donde MySQL tiene un porcentaje mayor de consumo que MongoDB, sin embargo donde sí se identifica una mayor ventaja en la gestión de recursos es el acceso a disco duro, donde el conteo de las operaciones de entrada y salida (E/S) son ampliamente superiores en el motor de bases de datos MySQL, por otra parte el rendimiento del procesador es relativamente similar en ambos motores presentándose una pequeña ventaja en MySQL sobre MongoDB.

En cuanto a los resultados de las pruebas de inserción, actualización, eliminación, búsqueda y funciones de agregado se puede decir de acuerdo a los tiempos arrojados por el sistema que MongoDB es más rápido en operaciones de inserción sobre la base de datos, ampliamente superior a los tiempos de respuesta de MySQL y muy similar los rendimientos de ambos motores en las operaciones de actualización, eliminación y búsqueda siendo superior aun así MongoDB dado sus menores tiempos de respuesta, sin embargo en las funciones de agregado es clara la ventaja que tiene el motor de bases de datos MySQL frente a MongoDB esto dado que en NoSQL se carece de un esquema lo que por un lado permite añadir/eliminar campos libremente sin tener que definir previamente la estructura pero por el otro son mucho más complicadas la definición de este tipo de funciones. Modelos como MapReduce y Aggregation Framework se han creado para permitir la realización de este tipo de cálculos de agregación en forma parecida a como se hace en los modelos relacionales sin tener en la actualidad una madurez suficiente en estos entornos.

Recomendaciones

Es importante tener en cuenta que solo se realizó las pruebas de rendimiento a través de un solo usuario definido en el sistema, siendo interesante para la obtención de resultados el haber definido más de un usuario como se hace en un entorno real de operación.

Sin lugar a duda los resultados y conclusiones generadas a partir de este trabajo nos indican que se debe evaluar inicialmente el entorno de desarrollo que se pretende intervenir con una solución de bases de datos, es importante definir muy bien un listado de requerimientos y casos de estudio que permitan desarrollar un adecuado diseño de la base de datos.

Trabajo Futuro

Con este trabajo de investigación se abre la posibilidad a realizar nuevos estudios, mejoras y aplicaciones que permitan realizar comparaciones de rendimiento más amplias donde las estructuras de bases de datos para ambos motores este compuesta por una mayor cantidad de tablas o colecciones si hablamos desde un lenguaje NoSQL.

Si lo que se desea es evidenciar que tan rápido es un motor de bases de datos frente al otro, en ciertas operaciones transaccionales, definitivamente los resultados de este trabajo de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

investigación nos brinda unas bases importantes frente a este objetivo, sin embargo para futuros trabajos se puede ampliar los objetivos de desarrollo centrándolos en la creación de consultas mucho más complejas como triggers o la implementación del modelo MapReduce de NoSQL para funciones de agregado.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

Ciges Regueiro, J.M, (2012). An Open Source NoSQL Solution for Internet Access Logs Analysis

Leite, G.S, (2010). Análise Comparativa do Teorema CAP Entre Bancos de Dados NoSQL e Bancos de Dados Relacionais

Brewer, E.A, (2000). You can have at most two these properties for any shared-data system. Traducción propia.

Codd, E.F, (1970). A Relational Model of Data for Large Shared Data Banks

Jakob Mattsson, A.L, (2010). Investigating storage solutions for large data - A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data. Traducción propia.

Córdova Espinoza, R.F, Cuzco Sarango, B.E, (2013). Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales.

Stonebraker, M. (2007). The End of an Architectural Era

Brito Zhunio, D.M, (2011). Estudio del uso de MongoDB como alternativa a las bases de datos relacionales tradicionales en aplicaciones web que requieren rapidez de lectura/escritura de los datos almacenados.

Gracia del Busto, H. (2012) Bases de datos NoSQL, Revista Telem@tica Vol 11

Sánchez, J. (2004) Arquitectura del DBMS Oracle

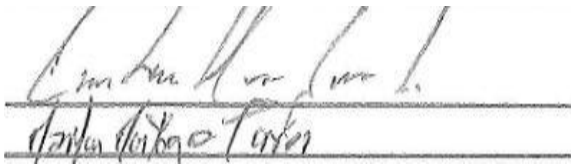
 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22


APÉNDICE

Tanto el producto final como la documentación del aplicativo que ayudó a la obtención de resultados se adjuntan en formato de DVD para consulta y soporte del lector interesado.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES





FIRMA ASESOR

FECHA ENTREGA: _____ 10 Agosto 2016 _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO____ ACEPTADO____ ACEPTADO CON MODIFICACIONES____

ACTA NO. _____
 FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____
 FECHA ENTREGA: _____