

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

COMPARACIÓN EXPERIMENTAL ENTRE SQL SERVER Y MONGODB

**DANIEL GOMEZ STUART
ÁLVARO ALEJANDRO AGUIRRE**

**ASESOR:
PEDRO ATENCIO ORTIZ M.SC.**

**TRABAJO DE GRADO PARA OPTAR AL TITULO DE TECNOLOGO EN
SISTEMAS DE INFORMACIÓN**

**INSTITUTO TECNOLOGICO METROPOLITANO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE SISTEMAS
2015**

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

RESUMEN

A medida que pasa el tiempo, la demanda de almacenamiento y velocidad de operaciones en las bases de datos se ha incrementado. Esto se debe a que más y más empresas tienen más aplicaciones, más clientes y quieren dar un mejor servicio. En algunas empresas la demanda de los servicios de bases de datos es tan elevada que no se pueden permitir hacer esperar al usuario, y como solución hasta el momento en algunas empresas se desea optimizar el sistema de información, tanto a nivel lógico como físico. Este proceso va desde verificar y optimizar las consultas de bases de datos hasta adquirir servidores con mejores especificaciones que permitan responder rápidamente a la demanda.

Como otra solución, surgió una nueva tecnología denominada No SQL, que se desprende del estándar por el que estaban regidos los anteriores motores de bases de datos, logrando mejores resultados en la realización de operaciones al motor como escritura y lectura de datos.

Este trabajo de grado pretende realizar una comparación experimental entre NoSQL y SQL, presentando los pros y contras de cada tecnología para que los IT de las empresas puedan elegir la opción que mejor se adapte a sus necesidades.

Para lograr esta propuesta, se realizan una serie de pruebas enfocadas en el rendimiento y la usabilidad de cada tecnología de bases de datos.

Palabras clave: Bases de datos, NoSQL, organizaciones, sistemas de gestión, empresas, servicios.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

RECONOCIMIENTOS

Son muchas las personas que colaboraron para que la elaboración de este proyecto fuese posible, sin embargo se da un agradecimiento especial al profesor Pedro Sandino Atencio Ortiz, quien deposito en nosotros toda su confianza para el correcto desarrollo de esta investigación.

·
 A nuestros padres porque gracias a su esfuerzo y dedicación logramos llegar a esta etapa de nuestras vidas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

ACRÓNIMOS

ACID: Acrónimo de Atomicity, Consistency, Isolation, Durability. Modelo que garantiza la atomicidad, consistencia, aislamiento y durabilidad de la base de datos. Esto es que todas las operaciones se deben realizar completamente o de lo contrario se regresa el sistema al estado antes del cambio; los datos a modificar sean los correctos, todas las operaciones sean independientes entre sí y los cambios realizados exitosamente perduren.

ANS: Acrónimo de Acuerdo de Nivel de Servicio (SLA en inglés). Contrato con el cliente sobre algunas características del sistema como disponibilidad, tiempos de respuesta, rendimiento, etc.

API: Traducido como Interfaz de Programación de Aplicaciones, son librerías para ser utilizadas por otras aplicaciones o servicios que permiten el uso de métodos, funciones, procedimientos, etc.

AWS (Amazon Web Services): Conjunto completo de servicios de infraestructuras y aplicaciones que permiten ejecutar prácticamente todo en la nube. (Amazon.com Inc.)

Back up: Respaldo de información.

BASE: Acrónimo de Basically Available, Soft state, Eventual consistency. Modelo alternativo a ACID que es básicamente disponible, de estado ligero y eventualmente consistente, es decir que no asegura la disponibilidad, el estado del sistema puede cambiar eventualmente incluso sin modificaciones y finalmente, el sistema llegará a un estado de consistencia con el tiempo mientras no reciba más ingresos.

BI: BusinessIntelligence: traducido como Inteligencia de Negocios, es un término que incluye aplicaciones, infraestructura, herramientas y las mejores prácticas que permiten el acceso y análisis de la información para mejorar y optimizar decisiones y rendimientos.

Column Families: Modelo avanzado de la estructura key-value. Similar a una tabla en un modelo de bases de datos relacional (RDBMS), contiene filas y columnas. Cada fila es identificada únicamente por una clave de fila; cada fila

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

tiene múltiples columnas, y cada una de estas tiene un nombre, valor y estampa de tiempo.

Confidencialidad: La información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.

DBA: Acrónimo de DataBase Administrator (Administrador de Bases de Datos). Persona encargada de realizar el mantenimiento y aseguramiento de los datos en un sistema de gestión.

Disponibilidad: Acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos, entidades o procesos autorizados cuando lo requieran.

EIDE: Acrónimo de Enhanced Integrated Drive Electronics (Unidad Electronica Integrada

Mejorada). También conocidos como PATA (Parallel Advanced Technology Attachment),

La etiqueta en estas unidades corresponde a la alternativa de interfaz en la que involucra conectar el tablero de la UCP y la unidad. Formatear este tipo de unidades puede tomar un largo tiempo ya que su velocidad es reducida.

EMR: Amazon Elastic MapReduce, implementa Apache Hadoop y Apache Hive por medio de MapReduce para controlar la complejidad de las aplicaciones además se integra con otros servicios de AWS.

Escalabilidad: Capacidad de mejorar recursos para ofrecer una mejora en la capacidad de servicio. (MSDN)

ETL: traducido como Extracción, Transformación y Carga, es un proceso que permite manipular para facilitar el uso en diferentes sistemas.

Ext3: Extended 3, o tercer sistema de archivos extendido, versión mejorada de ext2. Es un sistema de archivos que trabaja con registros por diario y utiliza un árbol binario balanceado

Ext4: Extended 4, o cuarto sistema de archivos extendido, versión mejorada de ext3. Es un sistema de archivos con registro que soporta volúmenes de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

almacenamiento de hasta 1024 PiB, usa menos recursos del sistema y mejora la lectura, escritura y verificación de los archivos.

Framework: Traducido como marco de trabajo, espacio designado para que los usuarios interactúen con el sistema proveyendo ventajas de usabilidad y funcionalidad.

Graph database: Almacena datos en grafos, la estructura más genérica de las estructuras de datos, capaz de representar cualquier dato de forma altamente accesible. Se leen los grafos siguiendo las flechas direccionales alrededor del diagrama para formar sentencias.

(Neo Technology). Un grafo es una manera avanzada de ver el almacenamiento key-value, ya que este primero se forma cuando los valores entre ellos mismos están interconectados.

GUI: Graphic User Interface, traducido como interfaz gráfica de usuario, es una herramienta o marco de trabajo visual para la interacción con el usuario.

Integridad: Mantenimiento de la exactitud y completitud de la información y sus métodos de proceso.

IRC: Internet Relay Chat. Sistema para chatear de tipo cliente-servidor.

JSON: JavaScript Object Notation. Es un formato para almacenar e intercambiar datos, independiente del lenguaje de programación que se esté utilizando.

(JSON Tutorial)

JRE: Java Runtime Environment, traducido como ambiente de ejecución de Java, es necesario que se encuentre instalado en las máquinas para correr aplicaciones basadas en Java.

Key Value database: Modelo que se puede considerar como 2 columnas, una con el valor de la “clave” –conocido como id o identificador-y otra con el valor “real” al que referencia la llave.

Little-endian: Formato en el sistema operativo en el que el byte de menor peso se almacena en la dirección más baja de memoria y el byte de mayor peso en la más alta.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Madurez: experiencia y conocimiento que tiene una empresa.

Mantenibilidad: se define como “facilidad con la que un sistema de software o componente puede ser modificado para corregir fallos, mejorar el rendimiento u otros atributos o adaptarse a cambios en el entorno”.

MapReduce: Consta de 2 partes “map” y “reduce”. Esta divide los flujos de trabajo (map) para procesarlos en paralelo y recombinar los datos procesados en la solución final (reduce).

Normalización: Cuando se asocia a las bases de datos, es un término que se utiliza usualmente durante el diseño de software por medio del modelo entidad relación. Existen varios niveles o formas de la normalización para organizar los datos, permitir flexibilidad y evitar redundancias.

NoSQL: Definido como Not only SQL, es un tipo de sistema de gestión de bases de datos que pretende ser la siguiente generación sobre estas tecnologías.
Replicación: copiar los datos de una ubicación a otras ubicaciones para evitar perder información.

RUP: Rational Unified Process, traducido como Proceso Racional Unificado, es un proceso iterativo e incremental de buenas prácticas para el ciclo de vida del desarrollo de software, desde la planeación hasta despliegue.

SATAo Serial ATA: Acrónimo de Serial Advanced Technology Attachments (Componente de Tecnología Avanzada Serial). Era una tecnología de bus primordialmente diseñada para transferir datos hacia y desde el disco duro. Contiene 2 conectores separados, uno para datos y otro para la energía, aunque puede tener un tercero para las conexiones de energía para PATA. (Serial ATA Connector Pinout)

SCSI: Acrónimo de Small Computer System Interface (Interfaz de Sistema para Pequeños Computadores). Su funcionalidad es similar a los EIDEs ya que utilizan discos magnéticos rotatorios para escribir y almacenar datos en este. La mayor diferencia entre SCSI, EIDE y SATA es cuando el tipo de disco SCSI gira más rápido por lo que procesa y almacena los datos más rápidamente, sin embargo es más propenso a dañarse debido a la velocidad de rotación

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

SSD: Acrónimo de Solid State Drive (Unidad de Estado Sólido). Compuesto de parte estacionarias opuesto a otras unidades que generalmente incluyen discos magnéticos para rotar mientras respaldan los datos. Están equipados con semiconductores que realizan la tarea de rotar discos magnéticos para apilar los datos almacenados. Al no poseer partes móviles son menos susceptibles de dañarse, además la velocidad de procesamiento es más rápida que otras unidades.

Tabla: En bases de datos, es un elemento en la cual se organizan y almacenan los datos en celdas, los valores de estas celdas están asociados a una fila y a una columna.

UCP: Unidad Central de Procesamiento. Parte de un sistema informático que interpreta y procesa los datos generados por los programas informáticos.

Usabilidad: Según la IEEE (Institute of Electrical and Electronics Engineers, 1990), se define como “facilidad con la que un usuario puede aprender a operar, producir entradas e interpretar las salidas de un sistema o componente”.

Versión: cuando se le atribuye al software, se interpreta como el estado o etapa en la que se encuentra una aplicación o servicio prestado.

ZFS: Sistema de archivos que cambia la forma como los sistemas de archivos son administrados, es robusto, escalable y fácil de administrar.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

CONTENIDO

1. INTRODUCCIÓN.....	12
2. MARCO TEÓRICO.....	16
3. METODOLOGIA.....	20
4. RESULTADOS Y DISCUSIÓN.....	30
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	57
REFERENCIAS.....	60
APÉNDICE A.....	62

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Figuras

Figura 1 Tiempos de respuesta prueba inserción 4.1 35

Figura 2 Tiempos de respuesta prueba de selección 4.1 37

Figura 3 Tiempos de respuesta prueba actualización 4.1 39

Figura 4 Tiempos de respuesta prueba masiva inserción 4.2 42

Figura 5 Tiempos de respuesta prueba actualización masiva 4.2..... 45

Figura 6 Tiempos de respuesta selección con JOIN 4.3 47

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tablas

Tabla 1 Motores RDBMS	18
Tabla 2 Motores No SQL.....	20
Tabla 3 Resumen de características de bases de datos	25
Tabla 4 Características servidor	31
Tabla 5 Características software	31
Tabla 6 Descripción tabla importacionesplano	32
Tabla 7 Descripción tabla tb_encuestas.....	33
Tabla 8 Descripción tabla aduanas	33
Tabla 9 Descripción tabla Empresas.....	34
Tabla 10 Uso recursos prueba inserción 4.1	37
Tabla 11 Uso de recursos selección 4.1	39
Tabla 12 Uso recursos Actualización 4.1	40
Tabla 13 Uso recursos Inserción masiva 4.2.....	44
Tabla 14 Uso recursos actualización masiva 4.2	46
Tabla 15 Uso recursos selección con JOIN 4.3.....	49
Tabla 16 Complejidad creación consultas prueba a.....	50
Tabla 17 Complejidad creación consultas prueba b.....	53
Tabla 18 Complejidad creación consultas prueba c	54
Tabla 19 Tiempos de respuesta prueba inserción 4.1.....	63
Tabla 20 Tiempos de respuesta prueba de selección 4.1	64
Tabla 21 Tiempos de respuesta prueba actualización 4.1	65
Tabla 22 Tiempos de respuesta prueba masiva inserción 4.2	66
Tabla 23 Tiempos de respuesta prueba actualización masiva 4.2	67
Tabla 24 Tiempos de respuesta selección con JOIN 4.3	68

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

1. INTRODUCCIÓN

El continuo incremento en el uso del internet y las nuevas tecnologías representan un costo para los proveedores de servicios, por ejemplo, a medida que más usuarios visitan un sitio web, el administrador de este requiere más servidores y con más recursos para poder soportar la demanda y evitar que el sitio colapse. Es decir, en la medida que más usuarios hacen uso de un servicio en internet, este internamente realiza más transacciones ya que debe consultar en sus bases de datos toda la información correspondiente para poder entregársela al usuario en una forma comprensible.

Según la página web (Db-engines, 2014) en su ranking mensual, a nivel mundial los motores de bases de datos más usados son Oracle, MySql y Microsoft Sql Server. Estos tres motores de bases de datos siguen el paradigma relacional, es decir almacenan la información en diferentes tablas, y al momento de que el usuario realice una consulta esta información se une internamente para mostrar información organizada y lógica.

Según se indica en el artículo (NAHEMAN & WEI, 2014) las bases de datos relacionales presentan problemas en los siguientes escenarios:

1. Alta concurrencia en la lectura y escritura de datos
2. Acceso y almacenamiento masivo de datos

Alta escalabilidad y Disponibilidad. Básicamente lo que se debe entender al analizar estos tres escenarios es que en cualquier caso en el que una alta cantidad de usuarios accedan a la base de datos, ya sea para lectura o escritura de datos, esta se verá afectada, no necesariamente va a colapsar pero si se va a ver afectado en gran medida el rendimiento y la velocidad.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Como respuesta a esta problemática desde el año 2000 se han venido desarrollando una serie de alternativas a las bases de datos relacionales, llamadas bases de datos No-Sql.

Como explica Christoforos Hadjigeorgiou (Hadjigeorgiou, 2014), las bases de datos No-SQL son bases de datos planas, es decir no se requieren de relaciones entre las tablas, ya que cada registro es dinámico, no requiere de una estructura predefinida y puede tener tantos campos como se desee, por lo tanto cada registro contiene toda la información que se pueda requerir sobre este.

No-SQL es una tecnología muy joven y la documentación sobre esta es pobre y difícil de encontrar, sin embargo se pretende recopilar la información disponible para analizarla y poder comparar en un caso práctico las capacidades de No-SQL contra las de SQL.

En este trabajo se va a comparar MongoDB contra Microsoft SQL-Server en un caso específico en el que se requiere del manejo de un gran volumen de datos.

En la primera parte del proyecto se especifican los criterios de evaluación que se tomaran en cuenta al momento de realizar las pruebas comparativas, estos criterios se definirán según lo documentado en la literatura académica.

En la segunda parte se realizan las pruebas previamente definidas en los motores de bases de datos MS SQL Server y MongoDB. Todas las pruebas se realizan en múltiples ocasiones para obtener resultados más acertados y generar gráficas y promedios.

En la tercera parte se analizan los resultados obtenidos y con base en estos se sacan conclusiones.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Contexto

Este proyecto se realiza como requisito de trabajo de grado para obtener el título de tecnólogo en sistemas de información en el I.T.M. Se aplica un caso puntual de investigación y comparación de la empresa Asiscomex, en la cual se manejan grandes volúmenes de información.

Este proyecto se centra en las posibilidades de NoSQL y habla de los diferentes motores, sus fuertes y carencias y por supuesto los compara con su contraparte SQL. Esto con el propósito de llegar al personal académico principalmente a los estudiantes para de este modo mostrar el potencial de NoSQL abriéndoles nuevas puertas en el mundo académico y empresarial.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

3.1. OBJETIVOS

3.1 General

Realizar un comparativo experimental entre el motor de bases de datos relacionales SQL SERVER y el motor de bases de datos No-SQL Mongo BD a través de un caso de aplicación que requiere el manejo de grandes volumen de datos.

3.2 Específicos

Identificar los requerimientos de evaluación del caso específico sobre el que se va a trabajar, teniendo en cuenta los motores SQL Server y MongoDB

Diseñar las consultas en ambos motores para evaluar los requerimientos de evaluación identificados.

Realizar una comparación experimental entre ambos motores, utilizando las consultas definidas previamente.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

2. MARCO TEÓRICO

Bases de datos

Según DUSAN PETKOVIC (PETKOVIC, 2014), una base de datos es “Una colección de datos que están relacionados a nivel lógico”. Ampliando un poco esta definición podríamos decir que una base de datos es un conjunto de datos en forma de registros, organizados en una serie de tablas relacionadas entre sí.

Relational database management system (Rdbms)

Según Bryan Syverson (Syverson & Murach, 2014), “Las bases de datos relacionales fueron desarrolladas en 1970 por Dr E. F. Codd. Este tipo de base de datos elimina algunos de los problemas que estaban asociados con los archivos comunes y otros diseños de base de datos, usando este modelo se puede reducir la redundancia de los datos, lo que ahorra espacio de almacenamiento y conlleva a una recuperación de datos más eficiente. Una base de datos relacional está formada por tablas. Las tablas están formadas por filas y columnas las cuales también se pueden llamar campos y registros”.

El modelo relacional de base de datos básicamente es un paradigma consistente en que cada registro debe ser único, en el caso de que un registro deba repetirse, este debe colocarse en una tabla diferente y relacionarse mediante la llave primaria con el registro principal.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Motores rdbms más usados

En la Tabla 1 se listan los motores RDBMS más comunes

Tabla 1 Motores RDBMS

Motor	Descripción
Microsoft Sql server	Motor de bases de datos de Microsoft, su licencia debe ser pagada y solo funciona sobre sistemas operativos Windows
Oracle Database	Es el motor de basa de datos más popular del mundo, su licencia es de pago, funciona sobre Windows, Linux y Unix.
MySql	El motor de base de datos libre más popular, es gratuito para uso no comercial, funciona sobre Windows, Linux y Unix.
MariaDB	Basado en MySql surge como una necesidad de una versión del motor que no dependa de Oracle, su licencia es libre, Windows, Linux y Unix
PostgreSQL	El primer motor de bases de datos orientado a objetos, su licencia es libre, funciona en Windows, Linux y Unix

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

SQL

Según DUSAN PETKOVIC (PETKOVIC, 2014)

“SQL, Lenguaje de consulta estructurado, es un lenguaje orientado a conjuntos. Esto quiere decir que SQL puede consultar múltiples filas de una o varias tablas con una sola sentencia. Esta característica es una de las ventajas más importantes de SQL, permitiendo el uso de este lenguaje a un nivel lógico más alto del que se puede usar en los lenguajes tradicionales“.

Se puede resumir que SQL es un lenguaje de programación que a diferencia de los lenguajes convencionales permite consultar conjuntos de datos mediante una sola sentencia.

NoSql

No-SQL es un paradigma de bases de datos que pretende solucionar las falencias de las bases de datos relacionales, cambiando el concepto de tablas con esquemas predefinidos y relaciones entre sí, por bases de datos planas, es decir sin tablas relacionadas, cada registro contiene toda la información requerida, para lograr esto cada registro se puede crear dinámicamente sin un esquema predefinido, es decir, cada registro puede tener diferentes campos.

Tal como se explica en el artículo (Hadjigeorgiou, 2014), “Las bases de datos No-SQL no tienen un esquema predefinido, los registros pueden tener tantos campos diferentes como se requiera, lo que se puede definir como un esquema dinámico”

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Motores de NoSql más comunes

En la Tabla 2 se describen los motores NoSql más usados

Tabla 2 Motores No SQL

Motor	Descripción
CouchDB	Motor de NoSql libre, usa Json como formato de datos y JavaScript como lenguaje para consultas, funciona en Windows, Linux y unix
MongoDB	Motor de NoSql libre, usa Json como formato de datos y JavaScript como lenguaje para consultas, funciona en Windows, Linux y unix
Cassandra	Motor de NoSql concebido para funcionar en la nube, su licencia es libre pero ofrece soporte directamente por personal de Apache. funciona en Windows, Linux y unix
BigTable	Motor de NoSql Creado y utilizado por Google, creado para funcionar en pc económicos es decir no se requiere de un servidor potente para su funcionamiento.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

3. METODOLOGIA

3.1 Métricas de evaluación de bases de datos

Tanto MS SQL Server como Mongo-DB son motores de bases de datos. La funcionalidad de ambos es almacenar de forma organizada la información para que pueda ser consultada en un momento determinado. Aunque el objetivo de ambos motores es el mismo, el paradigma que manejan es muy diferente, desde la forma de almacenar los datos hasta el modo en que se consultan. Debido a esto una comparación entre los pros y contras de ambas tecnologías puede ser demasiado extensa. A continuación se reportan algunas de las métricas de evaluación más destacadas según una revisión de la literatura actual al respecto.

En el artículo (Nance, Losser, Iype, & Harmon, 2014) se habla principalmente de dos características, la seguridad de los datos y el rendimiento, NoSQL al almacenar los datos sin una estructura predefinida manejan un rendimiento muy superior al de SQL, un caso en el que se evidencia esto es la página web www.facebook.com la cual maneja millones de transacciones simultáneamente sin que el usuario note demora alguna. En cuanto a la seguridad las bases de datos NoSQL llevan una gran desventaja ya que la información no está cifrada y en la transferencia entre cliente y servidor los datos pueden llegar a ser interceptados por un pirata informático.

Una de las características más importantes una base de datos es la escalabilidad, la escalabilidad es simplemente la capacidad de un sistema para adaptarse a una necesidad sin que se vea afectada la calidad, como nos explican en el artículo (Khan, 2014), NoSQL presenta una ventaja con respecto a SQL debido a la fácil escalabilidad horizontal, lo que economiza tiempo, recursos y por supuesto dinero.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Una diferencia entre NoSQL y SQL que a simple vista no es muy importante pero que en realidad puede afectar mucho la operación de una empresa, es la popularidad de la tecnología de bases de datos, como se explica en el mencionado artículo (Khan, 2014), las bases de datos SQL son mucho más populares debido a que llevan mucho más tiempo en el mercado y han sido vendidas y promocionadas por grandes empresas tales como Oracle, Microsoft e IBM. La popularidad de un sistema de bases de datos afecta la funcionalidad de las mismas en el sentido de que si una tecnología de bases de datos es más popular, habrá más personal calificado para administrarla. Para el caso de SQL se enseña en la mayoría de las universidades y se usa en la mayoría de las empresas por lo que encontrar un trabajador calificado en el tema no presenta mayor problema para una empresa, mientras que con NoSQL puede ser muy complejo encontrar una persona con los conocimientos suficientes para dar una administración correcta a un servidor de bases de datos.

El concepto ACID, Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad, son un conjunto de características que se deben tener en cuenta para que una base de datos relacional funcione correctamente, cualquier base de datos SQL debe seguir estas características para que su funcionalidad sea óptima. En el artículo (Sharma & Dave, 2014) se explica cómo NoSQL no sigue este estándar, debido a que los datos se almacenan de forma plana, esto ayuda a que NoSQL tenga un mejor rendimiento, sin embargo hace que sea más complicada la minería de datos y la inteligencia de negocios.

Los lenguajes para hacer las consultas son totalmente diferentes, SQL tiene un lenguaje estándar, por lo general la diferencias entre un motor y otro son mínimas, y al momento de migrar de un motor de bases de datos a otro, el proceso es complejo solo si la base de datos lo es. En NoSQL por lo general cada motor maneja su propio lenguaje, debido a que no existe un estándar general, debido a esto una migración de un motor a otro puede ser tan complejo como crear la base

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

de datos desde cero, en el artículo (Strauch, 2014), se indica que lo más común en las bases de datos NoSQL es los datos se almacenen en formato JSON, este es el formato utilizado por JavaScript, por lo que es bastante conocido por los programadores, sin embargo en este las consultas tienden a ser más complejas que en SQL.

Otra diferencia muy importante entre las dos tecnologías mencionadas es la madures de las mismas, como se explica en el artículo (NAHEMAN & WEI, 2014), SQL es una tecnología muy madura, lleva muchos años corrigiendo errores y mejorando los algoritmos para que sean cada vez más óptimos, NoSQL en cambio es una tecnología muy inmadura, a pesar de que el concepto existe hace muchos años, fue apenas cerca del año 2000 que se empezó a tomar enserio, es por esto que a pesar de que hay muchos motores disponibles todos ellos están en fases muy iniciales y no son tan robustos como un sistema SQL. Debido a esto cuando se realiza un desarrollo utilizando SQL Server u otro sistema RDBMS es factible realizar desde el motor de base de datos todas las operaciones relacionadas con los datos almacenados ya que estos motores brindan una amplia serie de comandos y funciones que permiten realizar operaciones complejas directamente en la consulta. En No-SQL es muy diferente, debido a que los motores de este paradigma llevan muy poco tiempo en el mercado, las funciones que permiten realizar son muy limitadas, por esto al utilizar un motor de bases de datos No-SQL, lo más común es recuperar los datos en bruto y a partir de ellos realizar las funciones requeridas desde el entorno de programación.

Otra diferencia entre los paradigmas de bases de datos mencionados es la calidad en el soporte técnico, como se indica en el artículo (NAHEMAN & WEI, 2014). La calidad en el soporte técnico de SQL-Server y otros motores SQL es muy alta ya que la mayoría de las empresas que venden motores de SQL brindan soporte personalizado a cualquier usuario que lo pueda necesitar. Por el contrario en NO-SQL el soporte técnico depende más que todo de foros en los que el soporte viene

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

de la comunidad y no de la empresa desarrolladora del motor de bases de datos. Por este motivo el soporte no siempre es el mejor.

Después de haber analizado las diferencias entre los paradigmas de base de datos SQL y NO-SQL. Se han seleccionado las diferencias más significativas para el caso de uso que se analiza en este proyecto. Los criterios por los cuales se seleccionan estas características para ser comparadas son los siguientes:

- No todas las características pueden ser comparadas ya que algunas solo corresponden a uno de los dos paradigmas de bases de datos. Por ejemplo el concepto ACID solo se encuentra en SQL por lo tanto no se podría implementar en No-SQL para hacer una comparación del mismo.
- Para el caso de uso en particular que se estudia en este proyecto no es posible evaluar algunas características tales como la seguridad. Ya que para evaluar esta característica se requiere implementar una infraestructura de red lo que desviaría el proyecto de su propósito real.
- Algunas de las características no requieren una comparación ya que son un concepto claro para la comunidad académica y empresarial, por ejemplo la madures de cada paradigma, aunque es posible compararlos, ya está claro que SQL es mucho más maduro debido a que lleva mucho más tiempo en el mercado.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

A continuación en la Tabla 3 se resumen las características de bases de datos analizadas.

Tabla 3 Resumen de características de bases de datos

Característica	Descripción
Rendimiento y la velocidad para realizar consultas que no involucren JOINS	Para este caso definiremos rendimiento como la velocidad con la que se realiza una consulta y el consumo de recursos que produce en el servidor. En este punto se realizaran consultas sencillas y se analizara el comportamiento del servidor.
Inserción y actualización masiva de registros.	En este punto se verificara el rendimiento al insertar y modificar cantidades muy grandes de registros.
Recuperación de masiva de registros.	En este punto validaremos el rendimiento realizar consultas complejas que retornes muchos registros.
Complejidad en la creación de consultas.	En este punto verificaremos la complejidad de las consultas, es decir lo difícil que puede ser para un programador realizar una consulta tanto en MS SQL Server, como en Mongo-DB
Facilidad para la generación de estadísticas y la inteligencia de negocios.	Aquí se verificara que tan posible y que tan fácil es generar estadísticas relacionadas con la inteligencia de negocios.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

2.2 Diseño de consultas para la comparación de métricas

Rendimiento y la velocidad realizar consultas que no involucren JOINS.

Como indica Microsoft (Microsoft, 2014) al momento de diseñar una base de datos se debe tener certeza de que todas las operaciones importantes se realicen de forma rápida y correcta.

Por lo general cuanto mayor es la base de datos, mayores son los requisitos de hardware. Además otros factores pueden influir en el rendimiento de la base de datos tales como el número de sesiones o usuarios conectados.

Como se indica en el artículo (EAFIT, 2014), El tiempo de respuesta en las transacciones de una base de datos es un factor de productividad determinante en las empresas, es por esto que es de vital importancia poder medir esta variable y determinar si un determinado motor de bases de datos cumple con las necesidades de la empresa.

En esta investigación se realizarán pruebas con una base de datos de producción sobre tablas con millones de registros, se crearán consultas para MS SQL server y un equivalente a la misma consulta para Mongo-DB, en este apartado serán consultas sencillas que no tendrán ningún tipo de JOIN, ya que debido al paradigma que maneja NoSQL no permite el uso de JOINS y para probar consultas con JOINS se requiere reestructurar las tablas en NoSql, este caso se probará en un apartado diferente, se harán pruebas de consultas que implementen la selección de datos agrupados, datos ordenados, datos con un top y datos diferentes. Adicionalmente se realizarán inserciones en la tabla de datos digitados y datos provenientes de otras consultas. También se probará con la actualización de registros y la eliminación. Se tomarán mediciones del tiempo de respuesta en las transacciones y el consumo de recursos en el servidor en ambos motores de bases de datos. Resumiendo lo anterior las consultas en este apartado tendrán las siguientes características:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

- Las consultas se realizaran sobre una sola tabla(sin JOINS)
- Los consultas retornaran máximo 10000(diez mil) registros
- Se utilizaran los argumentos ALL, DISTINCT
- Se implementaran los modificadores GROUP BY

Ejemplos:

SQL Server:

- `SELECT * FROM clientes`
- `SELECT * FROM clientes WHERE nombre='corbeta'`

MongoDB:

- `db.clients.find()`
- `db.clients.find({name:"corbeta"})`

Inserción y actualización masiva de registros.

El rendimiento en la inserción y actualización de datos afecta directamente la operatividad de cualquier empresa especialmente si la concurrencia en la base de datos es alta. Como indica la universidad de Sevilla (Universidad de Sevilla, 2014), “Existen casos en los que las transacciones ejecutadas aisladamente originan nuevos estados consistentes, sin embargo las mismas transacciones ejecutadas concurrentemente pueden originar efectos como pérdidas de operaciones y/o violación de restricciones de integridad”. Extendiendo esta explicación el problema con la concurrencia se presenta cuando varios usuarios están actualizando un mismo registro, lo que ocasiona inconsistencias en la información ya que un usuario puede modificar un registro desconociendo las modificaciones que han realizado los demás usuarios del sistema. Para controlar esto se opta por permitir que solo un usuario pueda modificar un registro a la vez, esto ocasiona demoras en las transacciones de escritura.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Para comparar MS SQL Server y MongoDB en cuanto a operaciones de inserción y modificación masiva de registros se ingresaran y actualizaran más de cien mil registros simultáneamente desde varios equipos para verificar el rendimiento en ambos motores. En ese orden de ideas se las consultas de este apartado cumplirán estas paramétricas:

- Las sentencias UPDATE o INSERT deben modificar como mínimo cien mil registros.
- Se implementaran sub-consultas para la actualización e inserción de datos.
- No se utilizaran JOINS en las consultas ni en las sub consultas

Ejemplos:

MS SQL Server:

- `INSERT INTO NuevosClientes SELECT * FROM clientes`

MongoDB:

- `var c=db.clientes.find()
db.clientesnuevos.insert(c)`

Recuperación masiva de registros incluyendo el uso de JOINS

En este apartado se realizaran pruebas tanto sobre MS SQL Server como sobre verificar el tiempo de respuesta en ambos motores cuando se realizan consultas complejas que retornen millones de registros. Las consultas en MS SQL Server se realizan con multiples JOINS, LEFT JOIN, RIGHT JOIN, INNER JOIN. Para el caso de MongoDB no es posible realizar JOINS como tal ya que en el paradigma de NoSQL las tablas no son normalizadas (Hadjigeorgiou, 2014), es decir toda la información es almacenada de forma plana, por este motivo las consultas deben ser estructuradas de una forma muy diferente. Para dar un ejemplo de las diferentes formas en las que se almacena la información en SQL Server y MongoDB supóngase una tabla que tiene la lista de contactos de los clientes, para

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

el caso de MS SQL Server se tendría una tabla con los contactos y esta debiese tener una identificación que pueda relacionarse con la tabla de empresas para así obtener toda la información que se requiera de la empresa relacionada. Para el caso de MongoDB la misma tabla de contactos en cada registro debe contener toda la información de la empresa relacionada. En este apartado se realizaran consultas con estas características:

- Las consultas deben retornar más de 100000 (cien mil) registros.
- Las consultas en MS Sql Server deben implementas uno más JOINS.
- Para realizar las consultas en MongoDB las tablas deben ser modificadas para que su estructura sea plana y poder ejecutar una consulta equivalente a la de MS SQL server.

MS SQL Server:

- `SELECT * FROM clientes c1 LEFT JOIN empresas em on c1.idEmpresa=em.idEmpresa`

MongoDB:

- `var c=db.clientesconempresa.find()`

Complejidad en la creación de consultas.

El propósito de este apartado poner a prueba lo difícil que puede ser para un programador experimentado realizar consultas en MS SQL Server y en MongoDB, se pondrán a prueba tres programadores cada uno de ellos con más de dos años de experiencia en bases de datos y se le encomendara a cada uno realizar una consulta en los dos motores de bases de datos mencionados. Para medir la complejidad en cada uno de los lenguajes se medirá el tiempo en el que cada desarrollador genera la consulta, en el caso de que el desarrollador no fuese capaz de realizar la consulta se tomaran los datos de la parte que pueda realizar para analizarla, adicionalmente se medirá la longitud de la consulta generada, esta medición se realizará en cantidad de caracteres, numero de palabras y cantidad

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

de palabra reservadas utilizadas. Los requisitos en este apartado son los siguientes:

- Los programadores que realizan las consultas tienen como mínimos dos años de experiencia en bases de datos.
- A cada desarrollador se la asignara un requerimiento deberá llevar a cabo tanto en MS SQL Server como en MongoDB.
- Los requerimientos incluirán selección, modificación e inserción de datos.

Ejemplo:

- Elaborar un reporte que indique cuantas operaciones ha tenido cada empresa en el año actual, los datos deben agruparse por mes y deben mostrar los siguientes datos: Mes, número de operaciones, Valor operaciones, Operación más costosa, Operador con más operaciones.

Facilidad para la generación de estadísticas y la inteligencia de negocios.

Para SQL existen una infinidad de herramientas para la generación de estadísticas y la inteligencia de negocios, incluso existe MS SQL Server business intelligence edition, una versión especializada en la generación de estadísticas e indicadores facilitando la gestión de la inteligencia de negocios en las empresas. Para el caso de NoSQL se hace mucho más complicado obtener indicadores y estadísticas de inteligencia de negocios debido al polimorfismo en las tablas y registros, como indica el Dr. Barry Devlin (Dr. Barry Devlin, 2014), la inteligencia de negocios está directamente relacionada con el paradigma SQL, sin embargo MongoDB al ser un motor de bases de datos orientada a documentos presenta algunas ventajas para la inteligencia de negocios, tales como su agilidad y flexibilidad superior. En este apartado se estudiara la posibilidad de generar estadísticas e inteligencia de negocios desde MongoDB y se comparara la calidad de la información generada contra estadísticas de inteligencia de negocios generados desde MS SQL Server.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

4. RESULTADOS Y DISCUSIÓN

Las pruebas se realizaran sobre un servidor AMD con características medianamente potentes. El sistema operativo empleado para las pruebas es MS Windows, esto se debe a que uno de los motores evaluados es MS Sql Server que solo corre en sistemas Windows. A continuación en la Tabla 4 se consignan las especificaciones técnicas del servidor.

Tabla 4 Características servidor

Característica	Valor
Procesador	AMD Phenom X4 9950
Frecuencia Procesador	2600 MHz
Tipo Memoria	Ddr3
Cantidad Memoria	5 GB
Tipo almacenamiento	Disco rígido SATA 7.200 RPM
Capacidad almacenamiento	2 TB

En cuanto a las versiones de software que se utilizan se optó por el software más moderno en la mayoría de las aplicaciones. El sistema operativo que se utiliza es Windows server 2008 que no es el más actualizado, sin embargo aún cuenta con soporte por parte de Microsoft por lo que se acomoda a este caso.

En la Tabla 5 se lista el software utilizado

Tabla 5 Características software

Aplicación	Versión
S.O.	MS Windows server 2008 r2 Enterprise x64
MS Sql Server	SQL server 2012 web edition
MongoDB	MongoDB 3.0 x64

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Para realizar las consultas a los motores de bases de datos se utiliza una aplicación web desarrollada en c#. Esta aplicación envía las consultas 100 veces a cada motor y toma el tiempo de respuesta en milisegundos, luego genera una tabla con los resultados la cual es copiada a Excel para generar gráficas.

Para realizar las pruebas comparativas se tomara como referencia una base de datos de producción de la empresa ASISCOMEX. El nombre de la base de datos es I.M. y corresponde a una aplicación web de inteligencia de mercados. La página web de la aplicación es <http://asiscomex-interna.vps.arkix.com/im/Default.aspx> . La mencionada base de datos pesa aproximadamente 50 TB por lo que ideal para hacer pruebas con grandes cantidades de información. Para las pruebas que se realizaran se tomaran como base 4 tablas de la base de datos:

Importacionesplano: esta tabla tiene 150 columnas y 11879604 de registros, es una tabla muy grade con muchas relaciones a otras tablas. Adicionalmente tiene columnas de los tipos de datos más comunes por lo que es ideal para hacer pruebas de rendimiento. En la Tabla 6 se relaciona cuantos campos tiene de cada tipo.

Tabla 6 Descripción tabla importacionesplano

Tipo de dato	Cantidad campos
bigint	24
decimal	26
int	12
int identity	1
smallint	9
tinyint	20
varchar	58

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

tb_encuestas: es una tabla con una gran cantidad de registros pero con pocos campos es ideal para hacer pruebas simples y verificar el comportamiento de ambos motores ante pequeñas cantidades de datos. En la Tabla 7 se relacionan sus campos.

Tabla 7 Descripción tabla tb_encuestas

Tipo de dato	Cantidad campos
bit	2
datetime	1
int	4
int identity	1
nvarchar	15
smalldatetime	3

Aduanas: Es una tabla muy pequeña solo se utilizara para hacer pruebas con JOINS ya que esta tabla se relaciona con **Importacionesplano**. La relación de su cantidad columnas por tipo se lista en la Tabla 8.

Tabla 8 Descripción tabla aduanas

Tipo de dato	Cantidad campos
int identity	1
nvarchar	3

Empresas: Al igual que la tabla anterior es muy pequeña y será utilizada en joins debido a su relación con la tabla **Importacionesplano**. En la Tabla 9 se describe su estructura

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 9 Descripción tabla Empresas

Tipo de dato	Cantidad campos
int	2
int identity	1
nvarchar	10
varchar	1

4.1. Rendimiento y la velocidad para realizar consultas que no involucren JOINS

Para probar la velocidad de inserción se empleará la tabla tb_encuestas.

Se toman 1000 (mil) registros de la tabla de producción, se generan las sentencias INSERT y se genera la consulta en ambos motores mediante aplicación web creada para esta investigación. En total se almacenarán 100000 (cien mil) registros ya que como se explicó anteriormente la prueba se hará 100 veces.

La primera sentencia del INSERT es la siguiente:

SQL Server:

```

INSERT INTO tb_encuestas
(idencuesta, idEncuestador, Estado_Llamada, Contacto, NitEmpresa, Valor
Agregado, Satisfecho, Porque, Fecha_Encuesta, CambioAgencia, CambioCarga,
Observacion, CorreoContacto, RespuestaFinal, FechaLicitacion, Confir
mado, Declinado, Lllmar_Luego, ObservacionCarga, FechaConfDec, CargoCont
acto, idsegmento, nrooperimpo, nrooperexpo, motinferr, motnocontesta, )
VALUES ('14', '5', 'Contesta', 'Ivon Ramirez', '900306527', 'Buen
Precio', 'Si', '', '09/09/2014
15:32:00', 'No', 'No', null, 'rehcosa@gmail.com', 'No
Interesa', '12/04/2014 0:00:00', null, null, null, null, '12/04/2014
0:00:00', 'Coordinadora de comercio
exterior', '111', '0', '30', null, null, )

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

MongoDB:

```

db.tb_Encuestas.insert({
  'idencuesta':'14',
  'idEncuestador':'5'
  , 'Estado_Llamada':'Contesta'
  , 'Contacto':'Ivon Ramirez',
  'NitEmpresa':'900306527'
  , 'ValorAgregado':'Buen Precio'
  , 'Satisfecho':'Si'
  , 'Porque':''
  , 'Fecha_Encuesta':'09/09/2014 15:32:00'
  , 'CambioAgencia':'No'
  , 'CambioCarga':'No'
  , 'Observacion':''
  , 'CorreoContacto':'rehcosa@gmail.com'
  , 'RespuestaFinal':'No Interesa'
  , 'FechaLicitacion':'12/04/2014 0:00:00'
  , 'Confirmado':''
  , 'Declinado':''
  , 'Llmar_Luego':''
  , 'ObservacionCarga':''
  , 'FechaConfDec':'12/04/2014 0:00:00'
  , 'CargoContacto':'Coordinadora de comercio exterior'
  , 'idsegmento':'111'
  , 'nrooperimpo':'0'
  , 'nrooperexpo':'30'
  , 'motinferr':''
  , 'motnocontesta':''
})

```

Los resultados obtenidos se muestran en la Figura 1:

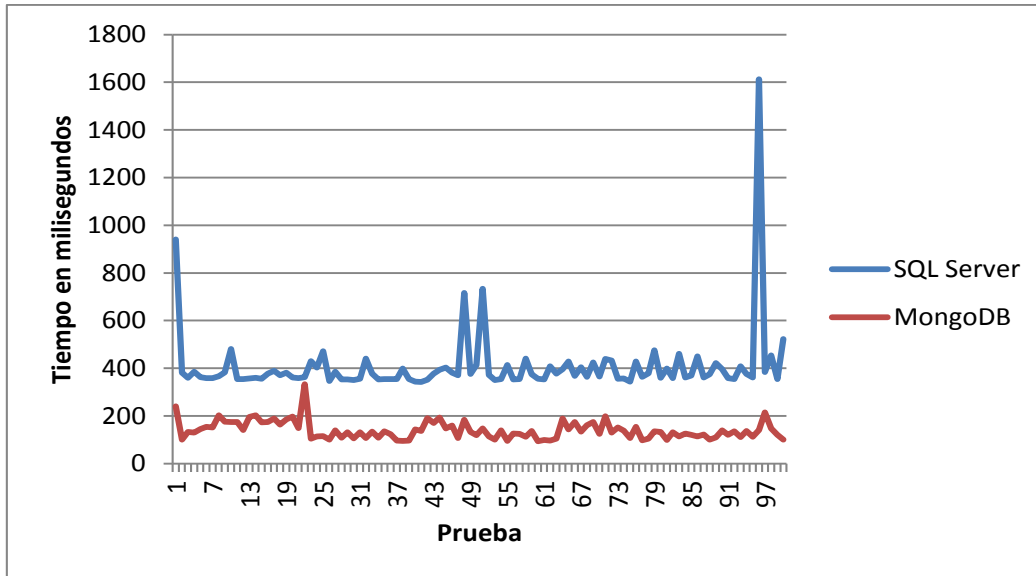


Figura 1 Tiempos de respuesta prueba inserción 4.1

El detalle de los resultados se encuentra en el apéndice A Tabla 19

En esta primera prueba se puede apreciar una leve diferencia entre los tiempos de ejecución de los dos motores, en MongoDB el tiempo de inserción fue en promedio de 139.72 milisegundos, menos de la mitad del promedio de SQL Server que fue 407.36 milisegundos. Esto indica que en cuanto a inserción de pequeñas cantidades de datos MongoDB tiene un mejor rendimiento que SQL Server.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

En la Tabla 10 se detalla el consumo de recursos durante la prueba

Tabla 10 Uso recursos prueba inserción 4.1

Uso de recursos		
	CPU	Memoria
MongoDB	2.8%	67%
SQL Server	3%	70%

En cuanto al consumo de recursos en el servidor, en ninguna de las pruebas se observó que se aumentara el consumo de CPU o de memoria RAM. La CPU se mantuvo al 3% y la memoria no supero el 70% de consumo.

Una vez se tienen los mismos datos tanto en Sql server como en mongoDB se procede con las pruebas de recuperación de registros. Para la medición se recuperarán todos los campos de los 100000 (cien mil) registros previamente almacenados.

La sentencia es la siguiente:

SQL Server:

```
SELECT * FROM tb_encuestas
```

MongoDB:

```
db.tb_Encuestas.find({})
```

Los resultados se muestran en la Figura 2:

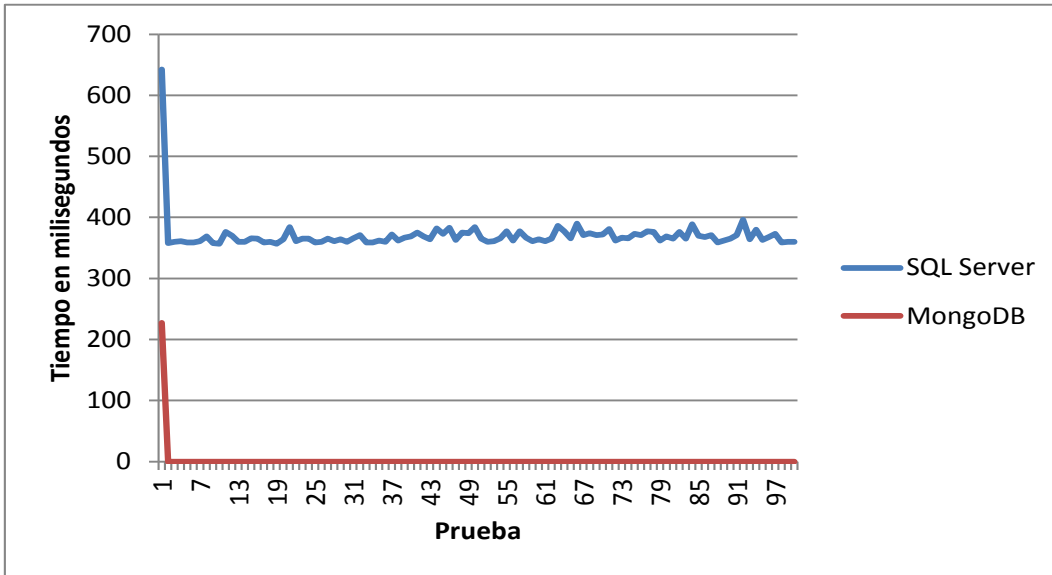


Figura 2 Tiempos de respuesta prueba de selección 4.1

El detalle de los resultados se encuentra en el apéndice A Tabla 20

En los resultados se observa un pico en la primera medición en ambos motores, sin embargo luego de esta se estabilizan los resultados, para SQL server el promedio fue de 370.35 milisegundos, mientras que para MongoDB fue de solo 2.27 milisegundos ya que en el 99% de las pruebas la respuesta fue instantánea y tomo 0 milisegundos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

A continuación en la Tabla 11 se detalla el consumo de recursos en la prueba

Tabla 11 Uso de recursos selección 4.1

Uso de recursos		
	CPU	Memoria
MongoDB	2.9%	70%
SQL Server	3.1%	72%

El consumo de recursos en el procesador y memoria fue imperceptible en la consulta realizada en ambos motores, los niveles se mantuvieron en el mismo margen que en la prueba anterior.

Por último para concluir con las pruebas sobre consultas sencillas se realiza un update a todos los registros de la tabla tb_encuestas, se modifica un campo de tipo texto, un campo numérico y un campo de fecha.

Las sentencias son las siguientes:

SQL Server:

```
UPDATE tb_encuestas SET idEncuestador=7, Contacto='Daniel Gomez', FechaLicitacion='20150101'
```

MongoDB:

```
db.tb_Encuestas.update(
  {
    'idEncuestador':7,
    'Contacto':'Daniel Gomez',
    'FechaLicitacion'='20150101'
  },
);
```

Los resultados se pueden observar en la Figura 3:

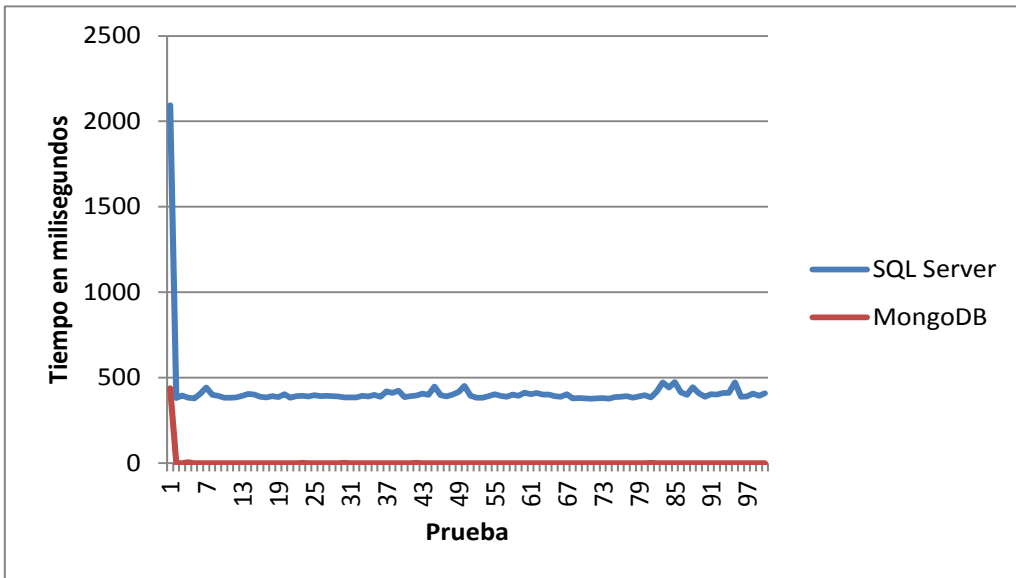


Figura 3 Tiempos de respuesta prueba actualización 4.1

El detalle de los resultados se encuentra en el apéndice A Tabla 21

El promedio de tiempo para SQL server fue de 415.78 milisegundo, mientras que para mongoDB el promedio fue de 4.48 milisegundos, al igual que en las pruebas anteriores el consumo de recursos en el equipo no se vio afectado.

En la Tabla 12 se puede observar el consumo de recursos en la prueba

Tabla 12 Uso recursos Actualización 4.1

Uso de recursos		
	CPU	Memoria
MongoDB	2.7%	72%
SQL Server	3.0%	72%

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

4.2 Inserción y actualización masiva de registros.

Para esta prueba se empleara la tabla importacionesplano, tanto en sqlServer como en MongoDB y se insertaran en ella 100000 (cien mil registros) en cada uno de los motores mencionados. Para mongoDB los datos se insertaran con una sentencia única, en Sql Server cada fila se insertará por separado ya que este motor solo permite insertar como máximo 1000 (mil) registros en un solo comando.

La prueba se realizará 100 veces por lo tanto se insertaran en total 10'000.000 (diez millones) de registros

Se necesitaron alrededor de 4 horas para completar esta prueba a continuación se anexan los resultados

Debido a que SQL server solo permite ingresar como máximo 1000 (mil) registros en una sola sentencia, en ese orden de ideas las sentencias de inserción son demasiado extensas, por este motivo solo se anexa a continuación parte de la primera sentencia.

SQL Server

```
INSERT INTO importacionesplano
(idImportacion,codigo_del_banco,aduana_presentacion,sucursal_del_banco,codigo_cajero,consecutivo_cajero,digito_verificacion,impo_exp_o,tipo_de_declaracion,numero_de_declaracion_anterior,fecha_de_la_declaracion_anterior,...) VALUES
('1','7','3','361','28','1748','1','','1','','null','575002871683','20111222','9520455652'...)
```


 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

MongoDB

Al igual que en SQL Server la sentencia en MongoDB puede ser muy extensa debido a la gran cantidad de datos, por este motivo en este documento solo se muestra parte de la sentencia para la inserción del primer registro.

```
db.importacionesplano.insert({
    'idImportacion':'1'
    , 'codigo_del_banco':'7'
    , 'aduana_presentacion':'3'
    , 'sucursal_del_banco':'361'
    , 'codigo_cajero':'28'
    , 'consecutivo_cajero':'1748'
    , 'digito_verificacion':'1'
    , 'impo_expo':''
    ...
})
```

En la Figura 4 se muestran los resultados obtenidos en la prueba

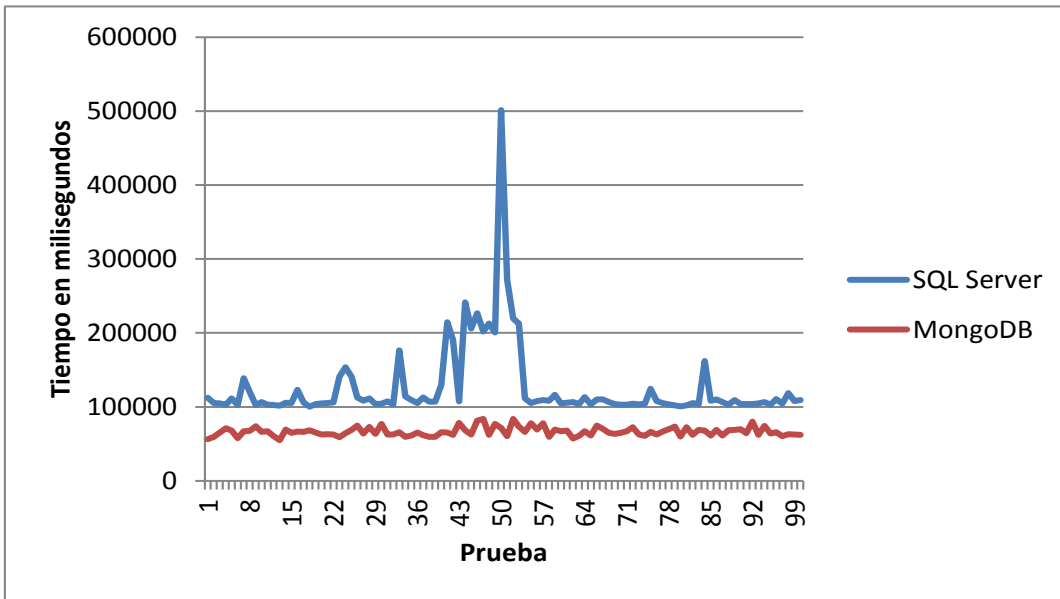


Figura 4 Tiempos de respuesta prueba masiva inserción 4.2

El detalle de los resultados se encuentra en el apéndice A Tabla 22

En promedio la inserción en SQL server tomo 126015.68 milisegundos un poco más de 2 minutos, en MongoDB el promedio fue de 66512.54 casi la mitad del tiempo que tomo en SQL Server.

Se pueden apreciar en la gráfica una gran cantidad de picos que se deben a que otros procesos estaban consumiendo recursos de servidor y debido a esto las consultas tienden a demorar más, sin embargo estos picos son mucho más pequeños en MongoDB que en SQL Server, lo que indica que MongoDB es mucho más estable y reacciona mejor ante el estrés del servidor.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

A continuación en la Tabla 13 se muestra el consumo de recursos de esta prueba

Tabla 13 Uso recursos Inserción masiva 4.2

Uso de recursos		
	CPU	Memoria
SQL Server	97%	90%
MongoDB	40%	87%

En cuanto al rendimiento y al uso de recursos para la Inserción. En SQL Server se vieron bastante afectados, el procesador durante la ejecución tuvo un promedio de uso del 97 por ciento y la memoria tuvo un promedio de uso de 90%.

Al realizar la inserción MongoDB los recursos del sistema también se afectaron bastante pero en una menor escala que con SQL Server el procesador tubo un uso promedio de 40%, el uso de memoria se mantuvo al 90%.

Para la prueba de Update se realizará una actualización a 10 campos de cada uno de los registros que se acaban de insertar a la tabla. Es decir se ejecutara en cada uno de los motores una sentencia de actualización sin ningún condicional para de este modo actualice todos los registros. Al igual que en la prueba anterior la sentencia se ejecutara 100 veces en cada uno de los motores para tener unos resultados más confiables.

Se modificaran datos de tipo numérico, Texto y Fecha. En este orden de ideas con esta sentencia de actualización de datos se probara el rendimiento de los motores con los tipos de datos más comunes.

La prueba se ejecutara simultáneamente desde tres equipos apuntando al mismo servidor para que así la prueba contemple las demoras de la concurrencia.

Las sentencias ejecutadas son las siguientes:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

SQL Server:

```
UPDATE importacionesplano SET
```

```
codigo_del_banco=7,aduana_presentacion=48,sucursal_del_banco=27,codigo_cajero=57,consecutivo_cajero=77,digito_verificacion=7,impo_expo=0,tipo_de_declaracion='Importacion',numero_de_declaracion_anterior='575002871683',fecha_de_la_declaracion_anterior='20150101',numero_de_manifiesto='48121212171717'
```

MongoDB

```
db.importacionesplano.update(
```

```
{
  'codigo_del_banco':7,
  'aduana_presentacion':48,
  'sucursal_del_banco':27,
  'codigo_cajero':57,'consecutivo_cajero':77,
  'digito_verificacion':7,
  'impo_expo':0,
  'tipo_de_declaracion':'Importacion',
  'numero_de_declaracion_anterior':'575002871683',
  'fecha_de_la_declaracion_anterior':'20150101',
  'numero_de_manifiesto':'48121212171717'
}
```

```
);
```

Los resultados de la prueba se muestran en la Figura 5:

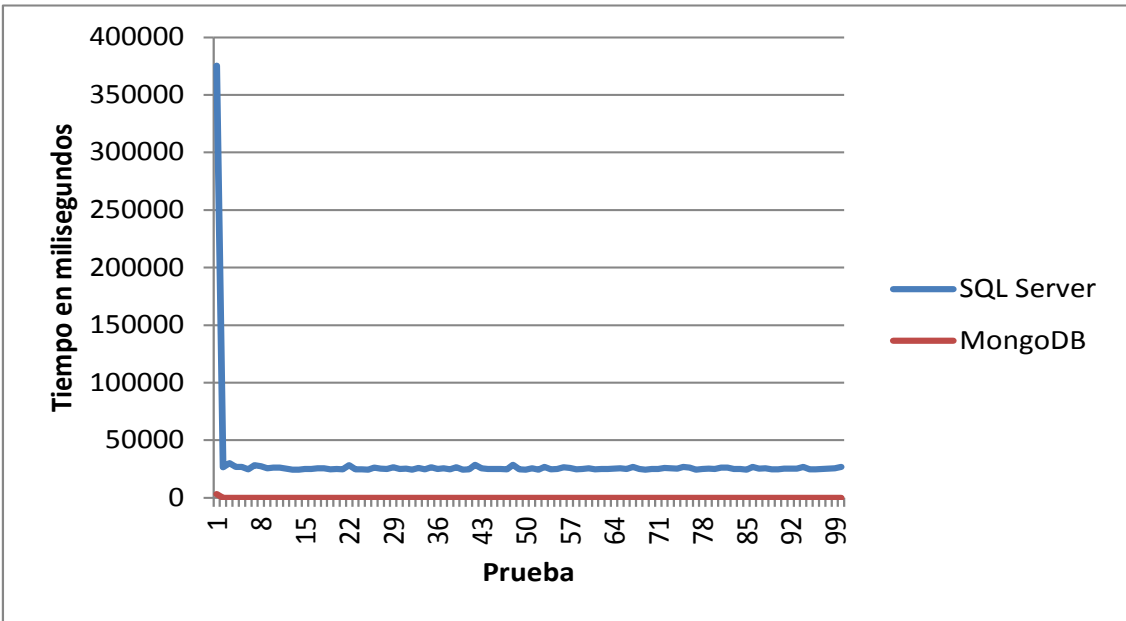


Figura 5 Tiempos de respuesta prueba actualización masiva 4.2

El detalle de los resultados se encuentra en el apéndice A Tabla 23
 El promedio de tiempo de ejecución para SQL Server fue de 29030.4 milisegundos mientras que para MongoDB fue de 32.91 milisegundos, el uso de recursos del equipo se puede observar en la Tabla 14

Tabla 14 Uso recursos actualización masiva 4.2

Uso de recursos		
	CPU	Memoria
SQL Server	45%	77%
MongoDB	3%	70%

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

En la prueba en SQL Server el procesador mantuvo un consumo promedio de 45% de procesador y 77% memoria, mientras que durante la ejecución de la sentencia en MongoDB el consumo de procesador no se vio afectado y se mantuvo al 3% y la memoria estuvo al 70% de uso.

4.3 Recuperación masiva de registros incluyendo el uso de joins

Para poner a prueba esta métrica se debe recrear un escenario diferente en cada motor de base de datos ya que MongoDB no permite el uso de joins de forma nativa. Para el caso de Ms SQL Server se emplearan las tablas Empresas y la tabla Aduanas las cuales harán join con la tabla importacionesplano y se recuperarán todos los campos de las 3 tablas.

Para el caso de MongoDB se debe modificar la colección importaciones planos y agregar a esta todos los campos de las tablas Empresas y Aduanas. Esto se realizara mediante un update a toda la colección.

Al igual que en las pruebas anteriores se realiza 100 veces la prueba sobre cada motor,

Las consultas ejecutadas son las siguientes:

SQL Server:

```
SELECT * FROM ImportacionesPlano ip left join aduanas ad ON
ip.aduana_= ad.idaduana left join empresas em ON
ip.nit__del_importador=em.identificacion
```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

MongoDB:

```
db.importacionesplano.find({});
```

Los resultados se muestran a continuación en la Figura 6:

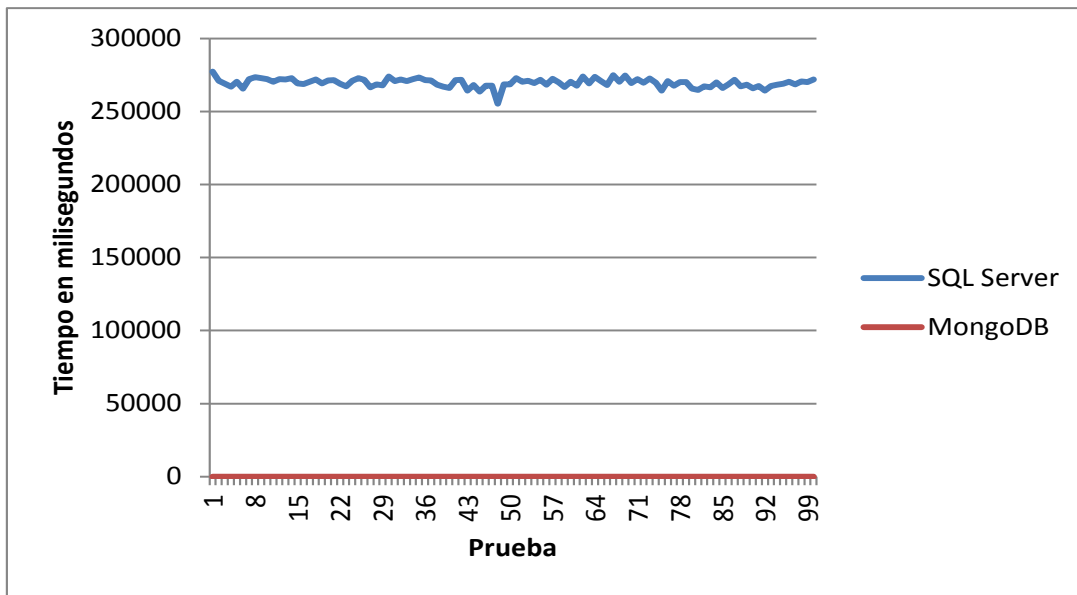


Figura 6 Tiempos de respuesta seleccion con JOIN 4.3

El detalle de los resultados se encuentra en el apéndice A Tabla 24

En la figura 6 se pueden observar los promedios en el tiempo de respuesta para SQL server y mongoDB, los cuales fueron de 269591.4 y 0.28, respectivamente. Sin embargo cabe destacar que me tomo 20 minutos crear la sentencia para agregar los datos correspondientes a Aduanas y Empresas a la colección importacionesplano. En este orden de ideas se puede decir que el rendimiento va a ser superior a largo plazo, pero para en el momento preciso se requiere un tiempo superior para la elaboración de consultas en las que intervengan datos de diferentes orígenes. A continuación en la Tabla 15 se muestran los resultados correspondientes al consumo de recursos durante las pruebas

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 15 Uso recursos selección con JOIN 4.3

Uso de recursos		
	CPU	Memoria
SQL Server	95%	97%
MongoDB	7%	72%

En cuanto al uso de recursos en esta prueba SQL Server llevo el servidor al límite, la CPU tubo un uso promedio de 95% y un uso promedio de memoria de 97%, mientras que en MongoDB el consume de recursos del procesador no se vio afectado.

4.4 Complejidad en la creación de consultas.

Para La realización de esta prueba se cuenta con la colaboración de 3 programadores entre los que figura Daniel Gomez Stuart uno de los creadores de este proyecto comparativo.

A continuación se detallan los requerimientos:

- a. Sobre la tabla importacionesplano realizar un reporte en el cual se indique la cantidad de operaciones que tubo cada empresa en el año 2014 ordenar la consulta por el número de operaciones descendentemente.
- b. Crear una tabla (SQL Server) o colección (MongoDB) con el nombre ImportacionTemporal en ella insertar el nombre de importador, la fecha del manifiesto y el número de aceptación de todas las operaciones de las empresas que hayan tenido más de 100 operaciones en el año 2014
- c. Construir una sentencia que actualice en la tabla ImportacionTemporal, la fecha del manifiesto de cada una de las operaciones en el caso de que se haya modificado.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Prueba a

El programador Numero 1 realiza las siguientes sentencias

Para SQL Server

```
SELECT razon_social_importador,COUNT(razon_social_importador) AS
NumeroOperaciones FROM importacionesplano WHERE
YEAR(CAST(fecha_del_manifiesto AS DATETIME))=2014 GROUP BY
razon_social_importador ORDER BY count(razon_social_importador)
DESC
```

Para MongoDB

```
db.importacionesplano.aggregate([
  $match:{fecha_del_manifiesto:/2014/}}
  ,{$group:({_id:"$razon_social_importador",count:{$sum:1}})}
  ,{$sort: {'count':-1}}
])
```

Las estadísticas de la prueba son las que se muestran en la Tabla 16

Tabla 16 Complejidad creación consultas prueba a

	SQL Server	MongoDB
Tiempo	1 Minutos	7 Minutos
Numero de caracteres	241	152
Cantidad Palabras	23	15
Palabras Reservadas	14	5

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

La sentencia de SQL Server es mucho más larga que la de MongoDB sin embargo el tiempo que le tomo al programador generarla es muy inferior al tiempo que le tomo generar su equivalente en MongoDB.

Prueba b

El Programador Numero 2 genera las sentencias a continuación:

SQL Server:

```
CREATE TABLE ImportacionTemporal (id INT PRIMARY KEY
IDENTITY(1,1), NombreImportador VARCHAR(500),numero_aceptacion
VARCHAR(20),fecha_del_manifiesto SMALLDATETIME)
INSERT INTO ImportacionTemporal
SELECT
razon_social_importador,numero_aceptacion,fecha_del_manifiesto
FROM #tmpimpo WHERE nit__del_importador in
(SELECT nit__del_importador FROM importacionesplano WHERE
YEAR(CAST(fecha_del_manifiesto AS DATETIME))=2014 GROUP BY
nit__del_importador,fecha_del_manifiesto HAVING
COUNT(nit__del_importador)>100)
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

MongoDB:

```

var tmp = db.importacionesplano.aggregate ([
    {$group: {_id:"$nit__del_importador",count:{$sum:1}}}
    ,{$sort: {'count':-1}}
    , {$match: {'count':{$gt : 100}}}
])
; var objNit=new Array;
for(var i=0;i<tmp.result.length;i++)
{
    objNit[i]=tmp.result[i]._id
};
var objInsert=db.importacionesplano.find({
    nit__del_importador:{$in:objNit},
    fecha_del_manifiesto:/2014/}
    ,{razon_social_importador:1
    ,numero_aceptacion:1,fecha_del_manifiesto:1
}).toArray();
db.ImportacionTemporal.insert(objInsert)

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

En la Tabla 17 se muestran las estadísticas de la prueba

Tabla 17 Complejidad creación consultas prueba b

	SQL Server	MongoDB
Tiempo	2 Minutos	17 Minutos
Numero de caracteres	495	467
Cantidad Palabras	45	56
Palabras Reservadas	25	18

La sentencia en SQL Server es un poco más larga sin embargo se debe tener en cuenta que el tiempo que tomo generarla al programador es mucho más corto que el tiempo que le tomo generar una sentencia equivalente en MongoDB.

Prueba c

El programador Numero 3 generas las siguientes sentencias:

SQL Server:

```
UPDATE it SET it.fecha_del_manifiesto=ip.fecha_del_manifiesto FROM
ImportacionTemporal it INNER JOIN importacionesplano ip ON
it.numero_aceptacion=ip.numero_aceptacion
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

MongoDB:

```

db.ImportacionTemporal.find().forEach(function(doc1)
{
var doc2=db.importacionesplano.find(
    {numero_aceptacion:doc1.numero_aceptacion}
    ,{fecha_del_manifiesto:1});
if(doc2!=null)
    {
        doc1.fecha_del_manifiesto=doc2.fecha_del_manifiesto;
        db.ImportacionTemporal.save(doc1);
    }
});

```

A continuación en la Tabla 18 se muestran los resultados obtenidos

Tabla 18 Complejidad creación consultas prueba c

	SQL Server	MongoDB
Tiempo	2 Minutos	150 Minutos
Numero de caracteres	168	264
Cantidad Palabras	19	27
Palabras Reservadas	6	8

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

En esta prueba la consulta de MongoDB no solo fue más larga en cantidad de caracteres y palabras, sino que también tomó mucho más tiempo construirla. Para la consulta en SQL Server el tiempo fue mínimo mientras que para MongoDB el desarrollador se demoró más de 2 horas. Aparte del tiempo esta fue la única consulta que el desarrollador no fue capaz de realizar solo con sus conocimientos ya que tuvo que recurrir a internet para consultar información al respecto en foros. La complejidad de esta consulta se debe a que en MongoDB no es posible hacer JOINS y dado que esta consulta enlaza dos tablas es necesario recurrir a JavaScript para hacer la relación lo que hace la consulta más compleja para el desarrollador además de que la hace más pesada para el servidor.

4.5 Facilidad para la generación de estadísticas y la inteligencia de negocios.

Para probar que tan posible y que tan fácil es hacer inteligencia de negocios desde MongoDB en comparación a SQL Server se plantearon un requerimiento el cual debe ser implementado en ambos motores, el requerimiento será desarrollado por un programador con más de 2 años de experiencia en ambos motores.

Requerimiento:

Se requiere saber cuáles empresas importadoras cumplen con un mínimo valor de importaciones mensual. Para esto tomar la sumatoria de el valor fob de las importaciones de cada empresa durante el año 2014 agrupando por mes e indicar cuales empresas cumplen y cuáles no. Tómese como valor mínimo \$ 10000 US (Diez mil dólares)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

SQL Server

```

SELECT nit_del_importador, razon_social_importador,
DATENAME(MONTH,MONTH(fecha_del_manifiesto)) AS [Mes],
SUM(valor_fob_US$) as [Total importaciones], CASE WHEN
SUM(valor_fob_US$)>10000 THEN 'Cumplido' ELSE 'Incumplido' END AS
[Indicador] FROM ImportacionesPlano WHERE
YEAR(fecha_del_manifiesto)=2014 GROUP BY
nit_del_importador,razon_social_importador,DATENAME(MONTH,MONTH(f
echa_del_manifiesto))

```

MongoDB

```

var Reporte=[];
db.importacionesplano.aggregate([
{
  $project:{mes:{$month:"$fecha_del_manifiesto"}
  ,nit:"$nit_del_importador"
  ,nombre:"$razon_social_importador"
  ,fob:"$valor_fob_US$"
  }
}
,{
  $match:{mes:10}
}
,{
  $group: {_id:{
    NitDelImportador:"$nit"
    ,RazonSocialImportador:"$nombre"
    ,Mes:"$mes"
  }
  ,total:{$sum:"$fob"},count:{$sum:1}}
}]).result.forEach(function(doc){
  if(doc.total>=10000)
  {
    Reporte.push(
    {
      "NitDelImportador":doc._id.NitDelImportador,

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

```

"RazonSocialImportador":doc._id.RazonSocialImportador,
    "Mes":doc._id.Mes,
    "TotalImportaciones":doc.total,
    "Indicador":"Cumplido"
    })
}
else
{
    Reporte.push(
    {
        "NitDelImportador":doc._id.NitDelImportador,
"RazonSocialImportador":doc._id.RazonSocialImportador,
        "Mes":doc._id.Mes,
        "TotalImportaciones":doc.total,
        "Indicador":"Incumplido"
    })
}
}); Reporte;

```

Como se puede observar en las consultas es tan posible hacer inteligencia de negocios e indicadores en un motor como en el otro sin embargo cabe destacar los siguientes puntos:

- El programador manifiesta mucha más dificultad para crear la consulta en MongoDB

- La consulta en MongoDB es más compleja y más extensa.

- En SQL Server el indicador se genera solo con código nativo de SQL, mientras que en MongoDB es necesario recurrir a JavaScript.

- A pesar de que es un indicador muy básico y en SQL Server fue construido en menos de 2 minutos, en MongoDB se necesitó más de una hora para completarlo.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES

- A lo largo del presente proyecto comparativo se ha conseguido detectar los pros y contras más representativos de los motores de bases de datos MS SQL Server y MongoDB al compararlos mutuamente.

Mediante las diferentes pruebas se ha conseguido demostrar que no es posible determinar superioridad de un motor frente al otro. Esto se debe a que a cada motor le corresponde un paradigma totalmente diferente, por lo que solo se puede catalogar un motor como mejor opción dependiendo del caso de uso específico para el que se pretenda utilizar.

- Debido a la amplia diferencia en el paradigma de los dos motores de bases de datos evaluados y con base en la información académica disponible en diferentes universidades se determina que los aspectos más relevantes a comparar son el rendimiento del motor de bases de datos tomando como base los tiempos de respuesta a solicitudes y en consumo de recursos, y la complejidad como tal en la sintaxis del motor

- Para una óptima veracidad de los resultados se crean consultas que evalúan cada uno de los motores, desde los casos más simples en cuanto a cantidad de datos y complejidad de la consulta, hasta los casos más complejos con enormes cantidades de datos y consultas complicadas. En este orden de ideas se crean consultas que tienen las operaciones básicas (Inserción, Modificación y selección) y se aumenta la complejidad de las mismas hasta llevar al límite cada uno de los motores evaluados.

- Desde la primera hasta la última prueba realizada demuestran el hecho de que MongoDB es mucho más rápido en cuanto a tiempo de respuesta a solicitudes, además de que es mucho más eficiente en el uso de recursos del sistema.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Especialmente en la realización de selecciones y modificaciones a los datos, se puede apreciar como MongoDB supera por mucho la velocidad de MS SQL Server. La debilidad de MongoDB en cuanto a SQL Server radica al momento de realizar consultas complejas. Mientras que el modelo relacional de MS SQL server permite realizar con rapidez consultas complejas, MongoDB requiere de mucho más código para obtener resultados similares, adicionalmente a esto el lenguaje SQL basta para realizar cualquier tipo de consulta que se plantee, mientras que el lenguaje de MongoDB no permite realizar muchas transacciones para las cuales es necesario complementar el código con JavaScript lo que hace que la consulta sea menos optima, más larga, y más difícil de entender por el desarrollados.

Teniendo en cuenta todos los resultados obtenidos de las pruebas se puede determinar que MongoDB es superior a SQL Server en cuanto a rendimiento en el manejo de grandes cantidades de datos, siempre y cuando la consulta sea sobre una sola tabla y no tenga que realizar operaciones complejas como en la inteligencia de negocios. En cualquier otro caso MS SQL Server sigue siendo una gran opción ya que gracias al modelo relacional permite que consultas con operaciones complejas se puedan realizar fácilmente y sin necesidad de utilizar otros lenguajes de programación que afecten el rendimiento de las mismas.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

5.2 Recomendaciones

- Pudieron haber sido más óptimos los resultados si se hubiesen tenido más recursos para la implementación de las pruebas, es decir todas las pruebas fueron realizadas sobre el mismo servidor y aunque se estima que el recrear las pruebas en un servidor diferente los resultados deben ser similares, hubiese sido ideal realizar las pruebas también en un servidor Intel y no solo en AMD.
- Las pruebas se realizan teniendo las bases de datos en un solo servidor por lo que en esta investigación no se prueban las capacidades de replicación o clustering.
- Se comparan solo un motor de base de datos relacional con un motor de bases de datos no relacionales, se puede mejorar la investigación probando múltiples motores de cada paradigma.

5.3 Trabajo futuro

Este proyecto abre las puertas para realizar comparaciones más amplias, no basándose solo en dos motores en particular si no comparando como tal el paradigma de bases de datos relacionales contra el paradigma no relacional, ya que existen muchos motores de bases de datos en cada uno de los paradigmas mencionados. También se puede complementar este trabajo comparando los diferentes motores de cada paradigma, es decir comparar los motores diferentes motores de NoSQL y aparte comparar los diferentes motores de SQL. Adicionalmente queda abierta la posibilidad de realizar las pruebas que se realizan en este proyecto en la nube, con bases de datos que se repliquen en múltiples servidores para de este modo calificar las capacidades de cada paradigma en diferentes ambientes y condiciones.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

REFERENCIAS

Db-engines. (2014). db-engines.

Dr. Barry Devlin. (2014). Business Intelligence NoSQL... No Problem. *10gen, Inc. / MongoDB*.

EAFIT. (2014). Evaluación del rendimiento de los motores de bases de datos Mysql y Firebird. *REVISTA Universidad EAFIT, Vol. 43. N.*

Hadjigeorgiou, C. (2014). *RDBMS vs NoSQL: Performance and Scaling Comparison*.

Khan, M.-A. (2014). *SQL vs NoSQL*. Retrieved from <http://www.cse.yorku.ca/~jarek/courses/6421/F12/>

Microsoft. (2014). Rendimiento de la base de datos. Retrieved from [http://technet.microsoft.com/es-es/library/ms190619\(v=sql.105\).aspx](http://technet.microsoft.com/es-es/library/ms190619(v=sql.105).aspx)

NAHEMAN, W., & WEI, J. (2014). Review of NoSQL Databases and Performance Testing on HBase. In *International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*.

Nance, C., Losser, T., Iype, R., & Harmon, G. (2014). *NoSQL vs RDBMS - Why is room for both*.

PETKOVIC, D. (2014). *Microsoft SQL Server 2008: A Beginner's Guide*.

Sharma, V., & Dave, M. (2014). SQL and NoSQL Databases. *International Journal of Advanced Research in Computer Science and Software Engineering*.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Strauch, C. (2014). *NoSQL Databases*.

Syverson, B., & Murach, J. (2014). *Murach's Sql server 2012 for developers*.

Universidad de Sevilla. (2014). Diseño de Bases de Datos Concurrencia. Retrieved from <http://www.lsi.us.es/docencia/get.php?id=1472>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

APÉNDICE A

Tabla 19 Tiempos de respuesta prueba inserción 4.1

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	941	240	35	355	136	68	364	161
2	382	100	36	354	123	69	425	174
3	360	133	37	355	96	70	365	124
4	386	130	38	399	95	71	439	198
5	362	145	39	355	96	72	433	130
6	359	154	40	344	143	73	356	151
7	358	152	41	343	137	74	357	138
8	367	202	42	352	191	75	344	107
9	383	176	43	378	171	76	429	154
10	481	174	44	393	193	77	364	98
11	354	174	45	403	147	78	379	105
12	354	141	46	382	160	79	476	136
13	357	196	47	370	107	80	360	133
14	360	202	48	716	184	81	399	99
15	356	173	49	376	133	82	359	131
16	378	175	50	414	119	83	460	114
17	389	189	51	734	148	84	361	126
18	370	163	52	372	114	85	369	121
19	381	185	53	351	101	86	450	114
20	361	197	54	354	139	87	361	122
21	359	149	55	413	95	88	374	100
22	363	333	56	353	126	89	422	110
23	430	104	57	355	125	90	396	140
24	403	114	58	440	112	91	359	120
25	471	115	59	375	137	92	355	136
26	346	100	60	356	94	93	408	111
27	386	140	61	353	99	94	376	137
28	353	108	62	408	96	95	361	112
29	353	131	63	377	105	96	1613	141
30	351	106	64	396	189	97	384	215
31	356	131	65	428	144	98	454	148
32	440	107	66	368	175	99	355	120
33	378	134	67	404	134	100	522	100
34	353	109						

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 20 Tiempos de respuesta prueba de selección 4.1

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	642	227	35	362	0	68	374	0
2	358	0	36	360	0	69	371	0
3	360	0	37	372	0	70	372	0
4	361	0	38	362	0	71	381	0
5	359	0	39	367	0	72	362	0
6	359	0	40	369	0	73	367	0
7	361	0	41	375	0	74	366	0
8	369	0	42	369	0	75	373	0
9	358	0	43	364	0	76	371	0
10	357	0	44	382	0	77	377	0
11	376	0	45	373	0	78	376	0
12	370	0	46	383	0	79	362	0
13	360	0	47	363	0	80	369	0
14	360	0	48	375	0	81	365	0
15	366	0	49	374	0	82	376	0
16	365	0	50	384	0	83	365	0
17	359	0	51	365	0	84	389	0
18	360	0	52	360	0	85	370	0
19	357	0	53	361	0	86	368	0
20	364	0	54	366	0	87	371	0
21	384	0	55	377	0	88	359	0
22	361	0	56	362	0	89	362	0
23	365	0	57	377	0	90	365	0
24	365	0	58	367	0	91	371	0
25	359	0	59	361	0	92	396	0
26	360	0	60	364	0	93	364	0
27	365	0	61	361	0	94	380	0
28	361	0	62	365	0	95	363	0
29	364	0	63	386	0	96	368	0
30	360	0	64	377	0	97	373	0
31	366	0	65	366	0	98	359	0
32	371	0	66	390	0	99	360	0
33	359	0	67	371	0	100	360	0
34	359	0						

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 21 Tiempos de respuesta prueba actualización 4.1

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	2094	438	35	399	0	68	379	0
2	380	0	36	388	0	69	381	0
3	396	0	37	419	0	70	379	0
4	382	6	38	410	0	71	377	0
5	379	0	39	424	0	72	379	0
6	404	0	40	386	0	73	381	0
7	441	0	41	392	0	74	377	0
8	399	0	42	395	1	75	386	0
9	393	0	43	407	0	76	387	0
10	383	0	44	399	0	77	392	0
11	383	0	45	447	0	78	382	0
12	384	0	46	398	0	79	389	0
13	393	0	47	389	0	80	397	0
14	405	0	48	401	0	81	384	1
15	401	0	49	415	0	82	417	0
16	388	0	50	451	0	83	471	0
17	384	0	51	394	0	84	441	0
18	391	0	52	383	0	85	473	0
19	386	0	53	383	0	86	414	0
20	402	0	54	391	0	87	399	0
21	382	0	55	402	0	88	443	0
22	391	0	56	393	0	89	408	0
23	394	1	57	387	0	90	388	0
24	389	0	58	400	0	91	402	0
25	398	0	59	394	0	92	400	0
26	392	0	60	412	0	93	411	0
27	393	0	61	403	0	94	410	0
28	391	0	62	410	0	95	471	0
29	390	0	63	400	0	96	387	0
30	384	1	64	400	0	97	390	0
31	384	0	65	391	0	98	406	0
32	384	0	66	387	0	99	394	0
33	393	0	67	402	0	100	409	0
34	389	0						

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 22 Tiempos de respuesta prueba masiva inserción 4.2

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	112312	56358	35	109080	61408	68	106656	64842
2	105242	59590	36	105242	65246	69	104030	63630
3	104636	65448	37	112716	61812	70	102818	64842
4	103020	70902	38	106858	59388	71	103020	66862
5	111302	68074	39	107060	59388	72	104232	72316
6	102414	57570	40	129684	65852	73	103222	62418
7	138572	67064	41	214322	65448	74	104434	61004
8	119988	67670	42	190082	62014	75	124432	66256
9	102616	73932	43	107464	78174	76	107868	62620
10	106454	66054	44	241188	68276	77	105242	66458
11	103020	66862	45	205838	62418	78	103222	69690
12	102414	60398	46	226442	81406	79	101808	73326
13	101606	54944	47	201798	83830	80	100596	59792
14	105646	69488	48	212908	62014	81	102010	72316
15	105040	64842	49	200384	76760	82	105242	62014
16	123220	66660	50	501364	71508	83	104030	68882
17	105848	66256	51	271690	60398	84	162206	67872
18	100192	68276	52	219372	83628	85	108474	61408
19	104030	65246	53	212706	73326	86	109888	69084
20	104838	62620	54	111302	66256	87	106656	61408
21	105242	63226	55	105242	77972	88	102818	68276
22	105848	62418	56	107666	69488	89	109282	68680
23	141198	59186	57	109282	77972	90	104030	69690
24	153520	64236	58	108272	59590	91	104030	64438
25	140592	69084	59	116352	69488	92	104030	79992
26	112716	74740	60	104636	66862	93	104636	62014
27	108474	63832	61	105646	67872	94	106656	74336
28	111504	73124	62	106656	57368	95	102818	63832
29	103626	63428	63	104030	60802	96	110696	65650
30	104434	77164	64	113120	67266	97	104232	60398
31	107464	62822	65	103222	61206	98	118776	63024
32	103222	62822	66	110292	74740	99	107666	62418
33	176548	65852	67	109888	70296	100	109080	62216
34	114130	59590						

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 23 Tiempos de respuesta prueba actualización masiva 4.2

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	375312	3215	35	26432	1	68	24864	3
2	26544	1	36	25088	2	69	24416	0
3	30016	0	37	25648	0	70	24976	3
4	26880	0	38	24752	0	71	24864	0
5	26880	1	39	26544	0	72	25984	0
6	24752	0	40	24528	2	73	25648	0
7	28336	1	41	24752	0	74	25200	3
8	27328	0	42	28448	3	75	26768	0
9	25536	0	43	25648	1	76	26208	3
10	26208	2	44	25088	0	77	24528	0
11	26208	0	45	25088	0	78	24864	0
12	25312	1	46	25088	1	79	25424	2
13	24528	0	47	24640	5	80	24864	2
14	24528	0	48	28672	0	81	26208	0
15	24864	2	49	24752	0	82	26208	0
16	24864	0	50	24416	3	83	24976	1
17	25648	0	51	25648	0	84	24864	0
18	25536	1	52	24304	4	85	24528	0
19	24752	2	53	26656	0	86	26656	0
20	25088	0	54	24640	0	87	25200	1
21	24640	0	55	24976	0	88	25648	0
22	28224	3	56	26544	0	89	24752	0
23	24752	0	57	25984	2	90	24752	0
24	24640	0	58	24640	3	91	25424	0
25	24528	0	59	24976	2	92	25312	0
26	26208	4	60	25648	0	93	25424	0
27	25200	0	61	24752	0	94	26880	0
28	24864	0	62	24976	2	95	24640	0
29	26544	3	63	25088	0	96	24752	0
30	24864	0	64	25424	0	97	24976	0
31	25424	3	65	25536	1	98	25424	0
32	24528	0	66	25088	0	99	25648	0
33	25872	2	67	26880	0	100	26768	0
34	24640	0						

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

Tabla 24 Tiempos de respuesta selección con JOIN 4.3

Resultado pruebas								
Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB	Prueba	SQL Server	MongoDB
1	277365	0	35	273183	0	68	270354	0
2	270969	0	36	271461	0	69	274536	0
3	268878	1	37	271215	0	70	269370	2
4	267033	0	38	268386	0	71	272076	1
5	270354	1	39	267033	0	72	269616	0
6	265680	0	40	266049	0	73	272568	0
7	272199	0	41	271338	1	74	269616	0
8	273429	0	42	271707	0	75	264327	0
9	272814	0	43	264204	0	76	270723	0
10	272076	0	44	268140	0	77	267525	0
11	270231	2	45	263589	0	78	270108	0
12	272076	1	46	267525	2	79	270108	0
13	271953	0	47	267648	0	80	265680	1
14	272814	0	48	255348	0	81	264696	0
15	269124	0	49	268509	0	82	267279	0
16	268755	2	50	268632	0	83	266418	2
17	270354	0	51	272691	1	84	269985	0
18	271953	0	52	270354	0	85	266049	0
19	269124	0	53	270969	0	86	268509	0
20	271215	0	54	269370	0	87	271707	0
21	271461	2	55	271707	0	88	267279	0
22	269001	0	56	268386	0	89	268386	1
23	267156	0	57	272322	2	90	265803	0
24	270969	0	58	269985	0	91	267402	0
25	272691	0	59	266664	0	92	264204	0
26	271707	1	60	270231	0	93	267402	0
27	266418	0	61	267648	1	94	268386	0
28	268632	0	62	273921	0	95	269001	0
29	267771	0	63	269247	0	96	270354	0
30	273798	0	64	273675	0	97	268632	2
31	270846	2	65	270723	0	98	270600	0
32	271953	0	66	268140	0	99	270108	0
33	270846	0	67	274782	0	100	271830	0
34	272076	0						



INFORME FINAL DE TRABAJO DE GRADO

Código	FDE 089
Versión	01
Fecha	2013-09-16

FIRMA ESTUDIANTES _____

FIRMA ASESOR _____

FECHA ENTREGA: _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO _____ ACEPTADO _____ ACEPTADO CON
MODIFICACIONES _____

ACTA NO. _____

FECHA ENTREGA: _____

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	01
		Fecha	2013-09-16

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____

FECHA ENTREGA: _____