



# **UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**

## **FACULTAD DE CIENCIAS DE LA INGENIERÍA**

### **CARRERA DE INGENIERIA EN SISTEMAS**

**Proyecto de investigación previo  
a la obtención del título de  
Ingeniero en Sistemas**

#### **TITULO DEL PROYECTO DE INVESTIGACIÓN:**

**“ESTUDIO COMPARATIVO ENTRE SQL Y NOSQL BASADO EN EL  
RENDIMIENTO, SEGURIDAD, E INTEGRIDAD DE LOS DATOS”**

#### **AUTOR:**

**MOREIRA RESABALA WILMER JOEL**

#### **DIRECTOR DEL PROYECTO:**

**LCDO. AMILKAR PURIS CÁCERES, PhD**

**QUEVEDO – LOS RÍOS – ECUADOR**

**2016**

## **DECLARACION DE AUTORIA**

El autor certifica que los criterios y opiniones vertidas en el presente trabajo de investigación, métodos y procedimientos utilizados en la información, análisis e interpretación de resultados son de exclusiva responsabilidad del mismo.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido en la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

---

**Firma**

**MOREIRA RESABALA WILMER JOEL**

## **CERTIFICADO DE CULMINACIÓN DE PROYECTO DE INVESTIGACIÓN**

Yo, **Lcdo. Amilkar Yudier Puris Cáceres, PhD**, Docente de la Facultad de Ciencias de la Ingeniería de la **Universidad Técnica Estatal de Quevedo**, certifico que el Sr **Wilmer Joel Moreira Resabala**, realizó el trabajo de investigación titulado **“Estudio comparativo entre SQL y NoSQL en aspectos de rendimiento, seguridad, e integridad de los datos”**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

---

Lcdo. Amilkar Yudier Puris Cáceres, PhD

**DIRECTOR DE TESIS**



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERIA**  
**CARRERA INGENIERÍA EN SISTEMAS**  
**PROYECTO DE INVESTIGACIÓN**

**Título:**

**“ESTUDIO COMPARATIVO ENTRE SQL Y NOSQL BASADO EN EL  
RENDIMIENTO, SEGURIDAD, E INTEGRIDAD DE LOS DATOS”**

Presentado al Honorable Consejo Directivo como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado por:

---

Ing. Byron Wladimir Oviedo Bayas

PRESIDENTE DEL TRIBUNAL DEL PROYECTO DE INVESTIGACIÓN

---

Ing. Pavel Novoa

MIEMBRO DEL TRIBUNAL

---

Ing. Efraín Díaz

MIEMBRO DEL TRIBUNAL

Quevedo – Los Ríos – Ecuador

AÑO 2016

## **AGRADECIMIENTO.**

A mis padres quienes con tanto sacrificio, constancia, amor y dedicación han forjado en mí una persona de bien, el mismo que les devolverá todo ese ofrecimiento que me han sido brindados a lo largo de los años.

También me gustaría agradecer incansablemente a mi tutor de trabajo de investigación, el Ing. Amilkar Puris Cáceres y al Ing. Ariosto Vicuña Pino, por su esfuerzo, paciencia y dedicación. Sus conocimientos, sus orientaciones, su manera de trabajar, su disponibilidad para trabajar y su motivación han sido fundamentales para mí.

Ellos han fomentado en mí un sentido de seriedad, responsabilidad y rigor académico sin los cuales no hubiese podido tener una formación completa como investigador. A su manera, han sido capaces de ganarse mi lealtad y admiración, así como sentirme en deuda con ellos por todo lo recibido durante esta fase que ha durado esta investigación.

## **DEDICATORIA.**

El presente tema de investigación lo dedico con todo mi afecto admiración y cariño a Dios, porque me dio la fe, y las fuerzas para no desmayar, la fortaleza necesaria para salir siempre adelante pese a las dificultades, por colocarme en el mejor camino, iluminando cada paso de mi vida, darme la salud y esperanza necesaria para dar por culminado este trabajo.

A mi familia especialmente a mis padres y novia quienes me han apoyado, gracias por su amor, esfuerzo y dedicación para seguir adelante vigorosamente en esta espinosa pero fructífera etapa.

## RESUMEN EJECUTIVO

A través del presente proyecto de investigación titulado **ESTUDIO COMPARATIVO ENTRE SQL Y NOSQL BASADO EN EL RENDIMIENTO, SEGURIDAD, E INTEGRIDAD DE LOS DATOS**, se exponen los fundamentos de los motores de bases de datos, tanto relacionales (SQL) como no relacionales (NoSQL), en función de sus indicadores, previamente planteados para su posterior comprobación.

Se realiza la investigación con el propósito de incrementar el nivel de conocimiento sobre bases de datos SQL y NoSQL en nuestro medio, el mismo que permitirá determinar el gestor de base de datos más viable de acuerdo a los indicadores planteados, debido a que estos métodos se complementan, es decir, se debe escoger a DBMS relacionales o no relacionales conforme a las circunstancias necesarias para su utilización.

Además se determina de qué manera se obtienen los tiempos de respuesta, al utilizar grandes volúmenes de información empleando un gestor de base de datos para almacenarlos, aplicando los indicadores basados en rendimiento, integridad y seguridad de los datos.

Al final se demuestra que sirve como una herramienta de apoyo, debido a que permite escoger a un gestor de base de datos ya sea del tipo relacional o no relacional, gracias a las ventajas y desventajas encontradas al momento de trabajar con una BD en aspectos de los indicadores antes mencionados.

**Palabras claves:** gestores, volúmenes, almacenamiento, monitoreo, análisis.

## ABSTRACT

Through this research project entitled “Comparative study between SQL and NOSQL BASED ON PERFORMANCE, SECURITY, AND INTEGRITY OF DATA”, fundamentals engine databases, both relational (SQL) and non-relational (NoSQL) are exposed depending on their indicators, previously set for later verification.

The research is done in order to increase the level of knowledge about SQL and NoSQL data in our midst, the same that will determine the database manager viable data according to the indicators proposed, because these methods are they complement, ie, must choose non-relational relational DBMS or under the circumstances necessary for use.

In addition, it determines how response times are obtained when using large volumes of information using a database manager to store, using indicators based on performance, integrity and security of data.

In the end it is shown that serves as a support tool, because it allows choose a manager database either relational or type non-relational, thanks to the advantages and disadvantages encountered when working with BD on aspects of the indicators above.

**Keywords:** managers, volumes, storage, monitoring, analysis.



## ÍNDICE DE CONTENIDO

DEDICATORIA.....	vi
RESUMEN EJECUTIVO.....	vii
ABSTRACT.....	viii
<b>INTRODUCCIÓN.....</b>	<b>14</b>
<b>1.1 PROBLEMA DE INVESTIGACIÓN .....</b>	<b>16</b>
1.1.1 Planteamiento del problema.....	16
1.1.2 Formulación del problema.....	16
Pronóstico del problema.....	16
<b>1.2 OBJETIVOS.....</b>	<b>17</b>
1.2.1 OBJETIVO GENERAL .....	17
1.2.2 OBJETIVOS ESPECÍFICOS .....	17
<b>1.3 JUSTIFICACIÓN .....</b>	<b>18</b>
<b>CAPÍTULO II. ....</b>	<b>19</b>
<b>FUNDAMENTACION TEÓRICA DE LA INVESTIGACIÓN .....</b>	<b>19</b>
2.1 MARCO CONCEPTUAL.....	20
2.1.1 Almacenamiento de Base de datos.....	20
2.1.2 Base de datos.....	20
2.1.3 BD NoSQL .....	21
2.1.4 BD SQL.....	21
2.1.5 Evolución de las bases de datos .....	21
2.1.6 MongoDB .....	21
2.1.7 MySQL.....	22
2.1.8 CRUD .....	22
2.1.9 Administrador de servicios de base de datos.....	22
2.1.10 Servicio de reportes .....	22
2.1.11 Tipos de bases de datos NoSQL.....	23
2.2 MARCO REFERENCIAL.....	24
2.2.1 Limitaciones para la base de datos SQL.....	25
2.2.2 Qué son bases de datos NoSQL.....	25
2.2.3 Ventajas de las bases de datos NoSQL.....	25
2.2.4 Quién usa MongoDB.....	25
2.2.5 JSON (JavaScript Object Notation).....	27
2.2.6 Comparación de los índices entre MongoDB y MySQL.....	28

2.2.7	Cuándo usar bases de datos SQL o relacionales. ....	28
2.2.8	SQL vs NoSQL: Diferencias de alto nivel. ....	31
2.2.9	Cuándo usar bases de datos NoSQL o no relacionales. ....	33
2.2.10	Especificaciones sobre NoSQL.....	33
<b>CAPÍTULO III .....</b>		<b>40</b>
<b>METODOLOGÍA DE LA INVESTIGACIÓN .....</b>		<b>40</b>
3.1.	Objeto de estudio realizado.....	41
3.2.	Tipo de Investigación.....	41
3.2.1.	Exploratoria.....	41
3.2.2.	Bibliográfica.....	41
3.3.	Método de investigación. ....	41
3.3.1.	Deductivo.....	41
3.3.2.	Analítico .....	41
3.3.3.	Sintético .....	41
3.4.	Fuentes de recopilación de información. ....	42
3.4.1.	Fuentes Primarias. ....	42
3.4.2.	Fuentes Secundarias.....	42
<b>CAPÍTULO IV.....</b>		<b>44</b>
<b>RESULTADOS Y DISCUSIÓN.....</b>		<b>44</b>
<b>4.2.RENDIMIENTO DE LOS GESTORES DE BASES DE DATOS.....</b>		<b>46</b>
4.5.	Primer prueba de rendimiento .....	51
4.6.	Segunda prueba de rendimiento .....	52
4.7.	Tercera prueba de rendimiento .....	53
4.8.	Cuarta prueba de rendimiento .....	54
4.9.	Resultados Gráficos de la Investigación. ....	55
<b>CAPÍTULO V.....</b>		<b>62</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>62</b>
<b>5.1CONCLUSIONES.....</b>		<b>63</b>
<b>5.2RECOMENDACIONES .....</b>		<b>64</b>
<b>CAPÍTULO VI.....</b>		<b>65</b>
<b>BIBLIOGRAFÍA.....</b>		<b>65</b>
<b>CAPÍTULO VII.....</b>		<b>69</b>
<b>ANEXOS .....</b>		<b>69</b>
7.2.	Proceso de instalación en NoSQL.....	81

## CONTENIDO DE TABLAS

Tabla 1. Principales comandos empleados en la consola MongoDB.....	26
Tabla 2. Principales parámetros empleados dentro de la consola MongoDB.....	27
Tabla 3. Diagrama de correspondencia entre entidades MongoDB y MySQL.....	30
Tabla 4. Materiales y recursos empleados para el desarrollo de la investigación.....	42
Tabla 5. Presupuesto del Proyecto de Investigación.....	43

## CONTENIDO DE GRÁFICOS

Gráfico 1. Representación gráfica en cuanto a SQL y NoSQL .....	20
Gráfico 2. Imagen tomada de <a href="http://mappinggis.com">http://mappinggis.com</a> .....	34
Gráfico 3. Representación gráfica de la comparación realizada entre gestores de base de datos SQL y NoSQL.....	47
Gráfico 4. Consola de Inicio en MySQL.....	51
Gráfico 5. Consola de Inicio en la consola MongoDB .....	51
Gráfico 6. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL .....	51
Gráfico 7. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB.....	52
Gráfico 8. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL .....	52
Gráfico 9. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB.....	53
Gráfico 10. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL .....	53
Gráfico 11. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB.....	53
Gráfico 12. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL .....	54
Gráfico 13. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB.....	54
Gráfico 14. Representación gráfica de la primera prueba realizada entre SQL y NoSQL .....	55
Gráfico 15. Representación gráfica de la segunda prueba realizada entre SQL y NoSQL.....	56
Gráfico 16. Representación gráfica de la Tercera prueba realizada entre SQL y NoSQL .....	57
Gráfico 17. Representación gráfica de la Cuarta prueba realizada entre SQL y NoSQL ..	57

Gráfico 18. Representación gráfica del total de los tiempos de respuesta obtenidos en aspecto de rendimiento seguridad e integridad de datos .....	49
Gráfico 19. En la gráfica que se presenta a continuación se puede apreciar los 10 DBMS más usados hasta Octubre del 2015.....	59
Gráfico 20. En la gráfica que se presenta a continuación se puede apreciar los 10 DBMS más usados hasta febrero del 2016. ....	59
Gráfico 21. Evolución de los gestores de base de datos.....	60
Gráfico 22. Cantidad de motores de base de datos por categoría.....	60
Gráfico 23. Popularidad de los motores de bases de datos.....	61
Gráfico 24. Preparación proceso de instalación de MySQL.....	70
Gráfico 25. Pantalla de Instalación de MySQL.....	71
Gráfico 26. Ventana de selección dl tipo de instalación en el equipo.....	71
Gráfico 27. Ventana para proceder a realizar la copia de archivos comunes .....	72
Gráfico 28. Ventana de selección para proceder a configurar MySQL.....	73
Gráfico 29. Ventana para seleccionar el tipo de configuración.....	73
Gráfico 30. Instalación de MySQL como servicio de Windows.....	74
Gráfico 31. Creación de nuevas credenciales de autenticación .....	75
Gráfico 32. Ejecución del proceso de instalación de MySQL como servicio de Windows	76
Gráfico 33. Instalación Completa de MySQL como servicio Windows .....	76
Gráfico 34. Visualización de MySQL como servicio de Windows.....	77
Gráfico 35. En la siguiente imagen se aprecia la consola MySQL.....	77
Gráfico 36. Pantalla de ingreso a la consola MySQL .....	78
Gráfico 37. Pantalla del Panel de Control. ....	79
Gráfico 38. Selección de Sistemas y Seguridad para configuración del Firewall .....	79
Gráfico 39. Selección de Características de Windows.....	80
Gráfico 40. Pantalla de activación de características de Windows .....	80
Gráfico 41. Visualización de MySQL como servicio de Windows.....	81
Gráfico 42. Menú Inicio comando cmd.exe .....	82
Gráfico 43. Consola en modo Administrador .....	82
Gráfico 44. Proceso de descarga del motor de base de datos MongoDB versión 3.0.7 ...	83
Gráfico 45. Carpeta destino de la instalación de MongoDB.....	83
Gráfico 46. Proceso de descarga del motor de base de datos MongoDB versión 3.0.7 ...	84
Gráfico 47. Instalación y configuración de MongoDB.....	84
Gráfico 48. Proceso para iniciar el servicio MongoDB .....	85

## CÓDIGO DUBLÍN

Titulo / Title	“Estudio comparativo entre SQL y NoSQL en aspecto de rendimiento, seguridad e integridad de datos”				
Autor / Creator	Wilmer Joel Moreira Resabala				
Materia / Subject	Facultad de Ciencias de la Ingeniería: Carrera Ingeniería en Sistemas				
Palabras claves / Keywords	gestores	volúmenes	almacenamiento	monitoreo	análisis
Descripción / Description	El siguiente proyecto tiene como objetivo, demostrar las ventajas que mantienen las bases de datos No relacionales, sobre las del tipo relacional. El mismo que consiste en un proyecto de investigación donde se determina cual es el Gestor de Bd más viable a la hora de trabajar con grandes volúmenes de datos.				
Editor / Publisher	Facultad de Ciencias de la Ingeniería: Carrera Ingeniería en Sistemas, Wilmer Joel Moreira Resabala				
Colaborador/ Contributor	Ing. Ariosto Vicuña Pino – Lcdo. Amilkar Puris Cáceres				
Fecha / Date	07 de Julio del 2015				
Tipo / Type	Proyecto de Investigación				
Formato / Format	Microsoft Word Office 2010				
Identificador / Identifier					
Fuente / Source					
Lenguaje / Language	Español				
Cobertura / Coverage	Quevedo – Los Ríos, Quito – Pichincha				
Derechos / Rights	UTEQ				
Audiencias / Audience	Proyecto de Investigación				

## INTRODUCCIÓN

En la actualidad y de manera general, para el almacenamiento de bases de datos existen dos tipos de gestores, el uno es SQL/Bd relacionales y el otro es NoSQL/Bd no relacionales, ambos son modelos que han venido contribuyendo de manera primordial con el desarrollo tecnológico, dichas bases de datos son utilizadas por casi todos los usuarios, porque se presentan muy prácticas, y además ofrecen beneficios al momento de almacenar registros.

Además, es importante no confundir términos usados exclusivamente en las bases de datos SQL y NoSQL, ya que son términos semejantes que hacen énfasis a su nombre, mas no al contenido que albergan cada uno de ellos, como por ejemplo: Las propiedades ACID, no se aplican en las NoSQL (no relacionales) por qué no existen relaciones, mientras que el teorema Brewer o teorema CAP no se aplica en las bases de datos relacionales (SQL) por qué es imposible para este gestor garantizar simultáneamente la consistencia, disponibilidad y tolerancia a fallos de los datos.

La investigación tiene como finalidad determinar las dificultades existentes entre ambos gestores, para ello utilizaremos los métodos deductivo, analítico y sintético que son la mejor opción para obtener datos objetivos, precisos y sistemáticos. Igualmente se tiene como propósito llegar a aquellas personas que desconocen cuáles son las similitudes, diferencias, ventajas y desventajas existentes entre estas herramientas, misma que nos ayudará a comprobar el gestor que sea de mayor viabilidad. Cabe recalcar que no hay que ver el tema como un "SQL vs NoSQL" sino como dos motores que mantienen ámbitos diferentes pero que conllevan a un mismo fin.

# **CAPÍTULO I**

## **CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN**

## **1.1 PROBLEMA DE INVESTIGACIÓN**

### **1.1.1 Planteamiento del problema**

Las instituciones alrededor del mundo y en la actualidad, se encuentran obligadas a trabajar con datos digitalizados a gran escala, para lo cual, se recopila esta información en BD relacionales, la misma que se encuentra alojada en uno o varios servidores propios de la empresa pero qué sucede cuando estos datos alcanzan los límites de almacenamiento de dichos servidores. Debido a la inexistencia de estrategias o herramientas que ayuden a mejorar el desempeño de las bases de datos, hace que el personal realice su trabajo en un entorno poco evolutivo, lo que indica que determinada empresa no estaría cumpliendo con sus objetivos planteados.

### **1.1.2 Formulación del problema.**

¿De qué manera se puede optimizar los tiempos de respuesta, enfocándose principalmente en los indicadores tales como rendimiento, seguridad e integridad al momento de utilizar un gestor de base de datos para conseguir almacenar grandes volúmenes de información vertiginosamente?

### **1.1.3 Diagnóstico del problema.**

- ✓ ¿De qué manera se podría conocer el rendimiento que mantienen los gestores de bases de datos relacionales y no relacionales?
- ✓ ¿Cuáles podrían ser las principales características que presentan las bases de datos SQL y las bd NoSQL?
- ✓ ¿Cómo se podrían puntualizar los componentes necesarios para comprobar la consistencia de los datos?

### **1.1.4 Pronóstico del problema.**

- ✓ Obtener resultados de desempeño entre modelos relacionales y no relacionales, a fin de conocer: los tiempos de respuesta que arrojaron ambos gestores de bases de datos, medir a sus indicadores, y determinar el gestor que sea de mayor viabilidad.



## **1.2 OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

Determinar los contextos de mejor desempeño de los motores de base de datos SQL y NoSQL, para así establecer cuál es el gestor de BD que permita optimizar los tiempos de respuesta independiente de las aplicaciones y/o páginas web utilizadas al manejar grandes volúmenes de datos.

### **1.2.2 OBJETIVOS ESPECÍFICOS**

- Evaluar el rendimiento de los gestores de bases de datos relacionales y no relacionales.
- Definir las ventajas y desventajas de los gestores de base de datos, utilizando MySQL para modelos relacionales y MongoDB para modelos no relacionales
- Establecer los mecanismos necesarios para poder garantizar la integridad de la información en ambos gestores de base de datos.

## 1.3 JUSTIFICACIÓN

Esta es una época en la cual estamos viviendo fuertes y constantes cambios tecnológicos, entre ellos el surgimiento de un nuevo modelo de almacenamiento de datos, llamado NoSQL/no relacionales<sup>1</sup>, el mismo que se encuentra captando la atención de los usuarios en esta área, gracias a su velocidad y eficiencia; para poder determinar los contextos de mejor desempeño de los motores de base de datos SQL y NoSQL se hace referencia al proceso realizado para analizar grandes conjuntos de datos de una complejidad de tipologías, en donde se logre obtener en el menor tiempo posible estos altos volúmenes de información.

Debido al incremento de información, se tiene la necesidad de establecer de manera detallada y comprobada el gestor más asequible. Para mantener un mejor rendimiento, mayor seguridad e integridad de los datos, se piensa ejecutar pruebas utilizando ambos gestores de almacenamiento, con el fin de salvaguardar un correcto cumplimiento de la base de datos, así como evitar pérdidas de tiempo y dinero.

La utilización de bases de datos no relacionales está siendo aplicada a varios proyectos de gran alcance, también en empresas muy importantes a nivel mundial tales como Google, Facebook, Amazon, twitter, entre otros; el mismo sirve para almacenar grandes volúmenes de datos, obteniendo logros como: mayor aplicabilidad, escalabilidad, alto rendimiento, y alta disponibilidad.

Éste contexto es de gran utilidad, pues debido a los beneficios que se van a alcanzar al determinar el desempeño obtenido de estos dos gestores de base de datos, permitirá al usuario incidir positivamente sobre dichos indicadores, así mismo se debe tener mucho cuidado a la hora de realizar comparaciones, no sólo porque sean diferentes, sino por la educación que se ha impartido a través de los años sobre base de datos en la carrera de sistemas.

---

<sup>1</sup> J. Fernández (2014), Bases de datos NoSQL I, Que es NoSQL. Quantummode.com

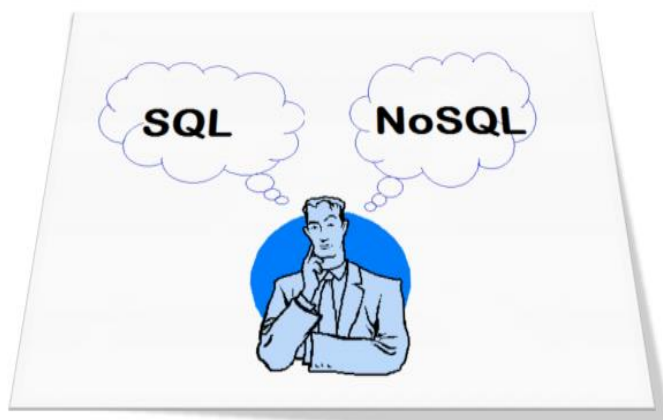
**CAPÍTULO II.**

**FUNDAMENTACION TEÓRICA DE LA  
INVESTIGACIÓN**

## 2.1 MARCO CONCEPTUAL.

A continuación se describen los principales conceptos, terminologías, y demás contenidos, relacionados con el proyecto de investigación vigente, en pocas palabras se presenta toda la información referente a estos dos modelos de almacenamiento de datos, para verificar sus fortalezas, debilidades<sup>2</sup>, ventajas, desventajas en diferentes áreas de la informática y de esta manera determinar el de mayor viabilidad. Como cita: “(Cordeiro 1998, p.50)...independientemente del nombre, estamos viviendo de las visiones que muchos analistas tengan sobre la era postmoderna, ciertamente el hombre en la actualidad protagoniza una nueva revolución: La Revolución de la Información”.

### Gráfico 1. Representación gráfica en cuanto a SQL y NoSQL



Fuente: Azharuddin, K. (2011)

**2.1.1 Almacenamiento de Base de datos.** Las bases de datos suelen ser creadas para almacenar grandes cantidades de datos de forma permanente<sup>3</sup>. Por lo general, los datos almacenados en éstas suelen ser consultados y actualizados constantemente. (Vera, 2006).<sup>4</sup>

**2.1.2 Base de datos.** Es un gran almacén que nos permite guardar información ordenada de tal manera que el usuario pueda leer e interpretar una gran cantidad de comandos dentro de la computadora. Una base de datos es un sistema de archivos electrónico.

---

<sup>2</sup> J. Fernández (2014), Bases de datos NoSQL IV, Las debilidades de NoSQL. Quantummode.com

<sup>3</sup> Leandro Alegsa (2011), Definición de Almacenamiento de bases de datos

<sup>4</sup> Aurelio López Ovando (2012), Almacenamiento de base de datos. aureliux.files.wordpress.com.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros.

**2.1.3 BD NoSQL.** Las bases de datos NoSQL, también llamadas No Solo SQL<sup>5</sup>, son un enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos.

NoSQL, que abarca una amplia gama de tecnologías y arquitecturas, busca resolver los problemas de escalabilidad y rendimiento de big data que las bases de datos relacionales no fueron diseñadas para abordar. NoSQL es especialmente útil cuando una empresa necesita acceder y analizar grandes cantidades de datos no estructurados o datos que se almacenan de forma remota en varios servidores virtuales en la nube. (Barragan, 2013).

**2.1.4 BD SQL.** SQL es el lenguaje más usado de momento para realizar el almacenamiento de información, es un lenguaje expresivo, el mismo que por defecto viene configurado en el idioma inglés.

**2.1.5 Evolución de las bases de datos.** El crecimiento o desarrollo de nuevas bases de datos es cosa del día a día dentro de todas las áreas conocidas hasta el momento, específicamente se habla de la evolución de los gestores de BD, ya que la base de datos tan solo determina la forma en la que será almacenada la información<sup>6</sup>.

**2.1.6 MongoDB.** Es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices

---

<sup>5</sup> Margaret Rouse (2015), No solo SQL (NoSQL). Barney Beal. Searchdatacenter.com

<sup>6</sup> Enrique Dans (2011), Entender el futuro: la evolución de las bases de datos. Universidad Santiago de Compostela. España.

secundarios, un lenguaje completo de búsquedas y consistencia estricta<sup>7</sup>.

Ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory). (Brito, 2011).

**2.1.7 MySQL.** Es sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Cuando tú tienes, por ejemplo, una página web, lo que tienes es un código que al abrir la URL de tu página web tu navegador lee y lo convierte en algo visual entendible para nosotros.

**2.1.8 CRUD.-** Acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Read, Update and Delete). Se usa para referirse a las funciones básicas en bases de datos<sup>8</sup>.

**2.1.9 Administrador de servicios de base de datos.** Es un tipo de aplicación que se ejecuta en el fondo del sistema.

**2.1.10 Servicio de reportes.** Es una arquitectura que admite diseñar, publicar y distribuir reportes que son generados de diferentes fuentes de datos.

- **Herramientas de Administración Básica.** Se utiliza para controlar y gestionar la base de datos así como para disminuir la cantidad de software instalado en el equipo.
- **Herramientas de Administración Completa.** A diferencia de la anterior se consigue todo es decir muestra las características y herramientas adicionales que se incluyen.

---

<sup>7</sup> Meera Prince (2014) MongoDB NoSQL Database / Document Database

<sup>8</sup> Nick Salloum (2015) CRUD (Create, Read, Update, Delete)

### 2.1.11 Tipos de bases de datos NoSQL.

Dentro de la arquitectura de almacenamiento de bases de datos no relacionales existen tres tipos considerados los más populares<sup>9</sup>.

❖ **Tiendas clave-valor (Key – value store).** Como su nombre lo indica, un almacén de claves-valor es un sistema que almacena los valores indexados para su recuperación por claves. Todos estos tipos de sistemas pueden contener datos estructurados o no estructurados.

❖ **Columna bases de datos orientadas (Column – Oriented database).** En lugar de almacenar conjuntos de información en una tabla fuertemente estructurada de columnas y filas con campos de tamaño uniforme para cada registro, como es el caso de bases de datos relacionales, las bases de datos orientadas a columnas contienen una columna extensible de datos estrechamente relacionadas.

❖ **Tiendas basadas en documentos (Document Store).** Estas bases de datos almacenan y organizan los datos como colecciones de documentos, en lugar de las tablas como estructuras con campos de tamaño uniforme para cada registro.

---

<sup>9</sup> Genbeta (2013) Bases de datos NoSQL. Elige la opción que mejor se adapte a tus necesidades, Mexico.

## 2.2 MARCO REFERENCIAL.

Dentro de la investigación realizada, tenemos varios contextos referentes al tema, los mismos que coadyuven con el desarrollo de este apartado. Uno de ellos es “Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales” el mismo que expresa una comparación coherente entre bases de datos relacionales y no relacionales.

“En la actualidad, las bases de datos relacionales se han proyectado a ser una de las herramientas más difundidas en nuestra sociedad de la información, las cuales permitan almacenar, manipular y recuperar información de diversos campos. No obstante, han tenido que pasar algunos años para que, estas bases de datos puedan alcanzar el desarrollo que hoy la conocemos, comenzando con el almacenamiento de información. Surge un modelo relacional, que un principio consistió en una serie de reglas para la evaluación de administradores de sistemas de datos relacionales.”

Podemos hacer referencia también al blog SQL vs. NoSQL ¿Cuál es el mejor?<sup>10</sup>. Al mismo que le daremos un enfoque diferente y analizaremos el por qué aparece la pregunta y en qué contextos es posible utilizar bases de datos relacionales, bases de datos no relacionales y plantearemos la posibilidad de realizar una especie de mezcla entre ambas tecnologías.

“Éste patrón es normal y a veces muy útil, permitiendo al lector sacar sus propias conclusiones y dejando que él mismo tome una decisión o postura al respecto, pero al tratar con tecnologías como SQL que son de facto en un desarrollo actual y NoSQL que desde su nombre es algo confuso, como explica Martin Fowler en su blog, debemos tener muchísimo cuidado a la hora de realizar comparaciones, no sólo porque sean diferentes, sino por la educación que se ha impartido a través de los años en todos los desarrolladores y estudiosos del campo de la informática.”<sup>11</sup>

---

<sup>10</sup> Genbeta (2013) Bases de datos NoSQL. Elige la opción que mejor se adapte a tus necesidades, Mexico.

<sup>11</sup> Ing. Esteban Zapata Rojas (2013), SQL vs. NoSQL ¿Cuál es el mejor?. [estebanz01.wordpress.com](http://estebanz01.wordpress.com)



### 2.2.1 Limitaciones para la base de datos SQL.

- **Escalabilidad:** Los usuarios tienen que escalar base de datos relacional en servidores poderosos que son caros y difíciles de manejar. Para escalar base de datos relacional que tiene que ser distribuido a varios servidores.
- **Complejidad:** En los datos del servidor SQL tiene que encajar en las mesas de todos modos.

Si los datos no se ajustan a las tablas, entonces necesita diseñar su estructura de base de datos que será complejo y más difícil de manejar.

### 2.2.2 Qué son bases de datos NoSQL.

- **No requiere esquema:** Los datos se pueden insertar en una base de datos NoSQL sin definir primero un esquema de base rígida.
- **Auto Elasticidad:** NoSQL se propaga automáticamente los datos en múltiples servidores sin necesidad de ayuda de la aplicación.
- **Caché integrada:** Con el fin de aumentar los datos a través de avance rendimiento NoSQL e incrementar los datos NoSQL usa técnicas de caché anticipadas desempeño en la memoria del sistema.

### 2.2.3 Ventajas de las bases de datos NoSQL.

- Las bases de datos NoSQL por lo general procesan datos más rápido que las bases de datos relacionales.
- Son también a menudo más rápido debido a que sus modelos de datos son más simples<sup>12</sup>.
- Los grandes sistemas de NoSQL son lo suficientemente flexibles para permitir una mejor utilidad para desarrollar aplicaciones de manera que respondan a sus necesidades.

### 2.2.4 Quién usa MongoDB

La lista de organizaciones que utiliza MongoDB es extensa, tenemos:

---

<sup>12</sup> J. Fernández (2014), Bases de datos NoSQL II, El modelo de datos. Quantummode.com

- ✓ Foursquare
- ✓ LinkedIn
- ✓ Empresas de telecomunicaciones como Orange y Telefónica.
- ✓ Empresas como Cisco, Bosch
- ✓ Plataformas de formación como Codecademy.
- ✓ eBay
- ✓ Expedia
- ✓ IBM
- ✓ Windows Azure
- ✓ McAfee
- ✓ Diarios como The Guardian, Le Figaro, The NewYork Times
- ✓ CERN (Organización Europea para la Investigación Nuclear) utiliza MongoDB para los grandes volúmenes de datos que genera el acelerador de partículas.

**Tabla 1. Principales comandos empleados en la consola MongoShell**

Comando	Descripción
<b>Insert</b>	Para insertar documentos en la colección.
<b>Find</b>	Para buscar o seleccionar documentos dentro de la colección.
<b>Count</b>	Para contar el total de documentos dentro de una colección.
<b>Update</b>	Para actualizar uno o varios documentos dentro de una colección
<b>Remove</b>	Para eliminar documentos de la colección.
<b>Drop</b>	Para eliminar una colección.
<b>Mongoexport</b>	Para exportar datos (en JSON o CSV). No debe usarse para backups completos.
<b>Mongoimport</b>	Para importar datos creados con mongoexport.
<b>Mongodump</b>	Para crear un export binario de los contenidos de una base de datos MongoDB. Debe usarse para backups.
<b>Mongorestore</b>	Para recrear una base de datos a partir de un dump creado con mongodump.

Fuente: Jam Blog 2014 - Isla de Margarita - Venezuela

De manera continua se describen los parámetros utilizados dentro de la Consola MongoDB Shell.

**Tabla 2. Principales parámetros empleados dentro de la consola MongoShell**

Parámetros	Descripción
<b>-h</b>	Este parámetro especifica el nombre de host (o servidor).
<b>--port</b>	Este parámetro especifica el puerto en el host que la base de datos MongoDB está escuchando.
<b>-d</b>	El nombre de su base de datos.
<b>-c</b>	El nombre de la colección para importar el archivo CSV.
<b>-u</b>	Desde la redacción requiere autenticar, el nombre de usuario de base de datos que ha especificado
<b>-p</b>	La contraseña de la base de datos que ha especificado
<b>---type</b>	El formato del archivo a importar, en general CSV o JSON.
<b>--file</b>	La ubicación física del archivo, en general en el equipo.
<b>--headerline</b>	Esto simplemente dice MongoDB hacer caso omiso de la primera línea del archivo CSV.

Fuente: López, D. (2014)

### 2.2.5 JSON (JavaScript Object Notation).

Es un formato empleado para el intercambios de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. Nació como una alternativa a XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa<sup>13</sup>. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación.

#### ¿Qué ventajas tiene la representación de nuestros datos utilizando JSON?

- ✓ JSON presenta un framework flexible y conciso tanto para consultas, como para almacenar registros.
- ✓ La sintaxis JSON es similar a la de otras estructuras de datos utilizados en muchos lenguajes de programación.
- ✓ JSON es independiente del lenguaje de programación que se utilice.

---

<sup>13</sup> Aurelio Morales (2014), Sobre MongoDB, bases de datos NoSQL y GIS. Formación GIS y difusión tecnológica, España.

### ¿Qué tipos de datos puede almacenar un JSON?

- Cadenas
- Números
- Textos booleanos
- Nulos
- Arrays
- Objetos/documentos

### 2.2.6 Comparación de los índices entre MongoDB y MySQL.

Un índice de base de datos se puede decir que es la estructura de datos que mejora la velocidad de las operaciones de recuperación de datos. Los índices son necesarios para las operaciones<sup>14</sup> de lectura de alto rendimiento para las consultas de uso frecuente. La exploración de base de datos completa es una operación muy lenta para ser utilizada en una gran cantidad de datos.<sup>15</sup>

**MySQL** utiliza índices en el nivel de tabla y soporta índices en cualquier columna de la tabla. Por otro lado, **MongoDB** utiliza índices en el nivel de colección y soporta índices en cualquier campo de los documentos en una colección MongoDB.

Como es bien sabido, la colección en MongoDB significa casi lo mismo que la tabla en MySQL. Lo mismo es bastante para los campos de MongoDB y columnas MySQL.

### 2.2.7 Cuándo usar bases de datos SQL o relacionales.

Gracias a las fuentes de información con las que he trabajado puedo considerar que una base de datos relacional puede ser utilizada en los siguientes medios<sup>16</sup>:

---

<sup>14</sup> José Manuel Alarcón (2014) Fundamentos de bases de datos SQL, operaciones con conjuntos, España.

<sup>15</sup> Javier Garzás (2013), ¿Bases de datos NoSQL o Bases de datos SQL? ¿Tiene sentido en “nuestro mundo” usar bases de datos NoSQL?. JavierGarzas.com

<sup>16</sup> Ana Milena Barragan Charry, Andrea Forero Sanabria (2013), implementación de una base de datos NoSQL para la Generación de la matriz o/d. Universidad Católica de Colombia, Bogotá. (Barragan, 2013)

- **Educativo:** es primordial conocer cómo organizar información, así como contribuir un alto conocimiento lógico al estudiante.
- **Desarrollo web:** Es favorable mantener la misma categoría en cuanto a los datos que entran, pero siempre y cuando la capacidad de concurrencia, almacenamiento y mantenimiento no mantengan una dificultad de consideración y que la información siempre sea consistente.
- **Rama de negocios:** La inteligencia de negocios, el análisis de negocios, las bodegas de datos, la minería de datos, y la minería de texto son temas que requieren el uso de SQL para facilitar la utilización de la información.
- **Empresarial:** Tanto programas a la medida como empresariales, ambos son de escritorio, y poseen la característica de mantener la información con una estructura consistente y SQL es una herramienta primordial para realizar ésta tarea.

A continuación, se puede ver el diagrama de correspondencia entre entidades MongoDB y MySQL.

**Tabla 3. Diagrama de correspondencia entre entidades MongoDB y MySQL.**

Descripción	MySQL	MongoDB 3.0.7
<b>Estructura Índice de datos</b>	B-Tree	B-Tree
<b>Punteros Índice</b>	Pila, índice agrupado	Archivos de memoria asignada
<b>Índice de la columna / campo de clave puede ser NULL</b>	Verdadero	Verdadero
<b>Índice agrupado (CI)</b>	SI	NO
<b>Los índices no agrupados (NCI)</b>	SI	SI
<b>Índice por defecto</b>	NO	SI
<b>Índices Compuestos / Integrado</b>	SI	SI
<b>Índices únicos</b>	SI	SI
<b>Índices espaciales / Geoespaciales</b>	SI	SI
<b>Índices de texto completo</b>	SI	SI
<b>Optimizador de consultas</b>	SI	SI
<b>Número máximo de índices por tabla / colección</b>	1000	64
<b>Número máximo de columnas / campos en el índice</b>	16	31
<b>Bytes por clave de índice</b>	900	1024

Fuente: Córdova, R. et al. (2013)

## 2.2.8 SQL vs NoSQL: Diferencias de alto nivel.

- ✓ SQL se las conoce como bases de datos relacionales (RDBMS); entretanto que NoSQL no son relacionales.<sup>17</sup>
- ✓ SQL son las bases de datos de la tabla, mientras que NoSQL son la colección de par clave-valor, documentos, bases de datos de gráficos o las tiendas de toda la columna que no tienen definiciones de esquema estándar que se debe respetarse<sup>18</sup>.
- ✓ SQL combina de manera eficaz distintas tablas para extraer información relacionada en cambio NoSQL no lo permite y en caso de que lo permita sería muy limitada.
- ✓ SQL permite escalar de manera complicada si no tiene un diseño distribuido mientras que NoSQL permite un escalado horizontal sin problemas.
- ✓ SQL permite la gestión de datos ligado con las relaciones que hay entre ellos, y en NoSQL no existe este tipo de utilidades.
- ✓ SQL utiliza SQL para la definición y manipulación de los datos, que es muy potente. En NoSQL, las consultas se centran en la colección de documentos. A veces también se llama como UQL.
- ✓ **Para consultas complejas:** SQL es adecuada para la realización de consulta intensiva, entretanto que NoSQL no lo es.

NoSQL no cuenta con interfaces estándar para efectuar consultas complejas, y las propias consultas que contiene no son tan eficaces como en SQL.

- ✓ **Para el tipo de datos a almacenar:** SQL no es mejor opción para el almacenamiento de datos jerárquica. Pero, NoSQL se adapta mejor para

---

<sup>17</sup> Ing. Esteban Zapata Rojas (2013), SQL vs. NoSQL ¿Cuál es el mejor?.  
estebanz01.wordpress.com

<sup>18</sup> Sergio Eduardo Mancilla Escobar (2013), uso de bases de datos NoSQL documentales para crear sitios web de alto rendimiento. Universidad de San Carlos de Guatemala, Guatemala.

el almacenamiento de datos jerárquica, ya que sigue la clave-valor, una forma de almacenar datos similares a los datos JSON.

- ✓ **Para escalabilidad:** SQL son verticalmente escalable. Aunque probada en muchos entornos productivos suele, por norma, ser inferior a las bases de datos NoSQL, escalar SQL suele significar añadir un servidor mucho más potente. Por otro lado, NoSQL son horizontalmente escalable. Se puede añadir unos servidores más cómodamente en su infraestructura para manipular el gran tráfico.
- ✓ **Por aplicación basada transaccional alto:** SQL es mejor opción para aplicaciones de servicio pesado de tipo transaccional, ya que es más estable y promete la atomicidad, así como la integridad de los datos. Aunque se puede usar NoSQL para transacciones propósito, todavía no es comparable y lo suficientemente estable, de alta carga y para aplicaciones transaccionales complejas.
- ✓ **Para obtener asistencia:** Excelente soporte están disponibles para toda la base de datos SQL de sus proveedores. También hay gran cantidad de consultas independientes que pueden ayudarle con SQL para unos despliegues a gran escala. Por otra parte NoSQL todavía tiene que contar con el apoyo de la comunidad, y de expertos externos limitados que se encuentren disponibles para que usted pueda configurar y desplegar sus desarrollos NoSQL a gran escala.
- ✓ **Para las propiedades:** SQL hace hincapié en las propiedades ACID<sup>19</sup> (atomicidad, coherencia, aislamiento y durabilidad), mientras que NoSQL sigue el teorema Brewer o teorema CAP (consistencia, disponibilidad y tolerancia a la partición)<sup>20</sup>.
- ✓ **Para los tipos de DB:** En un alto nivel, podemos clasificar SQL, ya sea como de código abierto o cerca de fuentes de proveedores comerciales.

---

<sup>19</sup> Leonardo D (2013) ACID en las bases de datos, Argentina

<sup>20</sup> Cecilio Álvarez Caules (2014 /31 /diciembre) NoSQL y el Teorema de CAP, <http://www.cantabriatic.com/>



### 2.2.9 Cuándo usar bases de datos NoSQL o no relacionales.

En la actualidad podemos obtener varias fuentes de información al respecto, por medio de lo cual es más sencillo orientarse a ésta nueva tecnología. Teniendo en cuenta que las tecnologías NoSQL pueden ser empleadas en los siguientes ambientes<sup>21</sup>:

- **Redes sociales:** Por medio de las redes sociales, nace esta tecnología, la misma que comenzó a desarrollarse y mostrarse útil en el campo de la informática y la estadística.
- **Desarrollo Web:** Creo mucho más adecuado el uso de éstas tecnologías, debido a la poca igualdad de la información que se encuentra en la Internet, sin embargo, es posible realizar éstos desarrollos con SQL.
- **Desarrollo Móvil:** Hoy en día, las organizaciones a nivel mundial están combatiendo contra un gran altercado conocido como BYOD, el mismo que se detalla más bien como un fenómeno general, es decir se trata de recopilar información, la misma que se obtiene siempre será diferente por más que uno desee estructurarla y mantenerla estática.
- **Cloud (XaaS):** el término XaaS (Everything as a service) en español “Todo como servicio” y todos los temas dependidos a la nube, con NoSQL pueden adaptarse casi a cualquier necesidad del cliente.

### 2.2.10 Especificaciones sobre NoSQL.

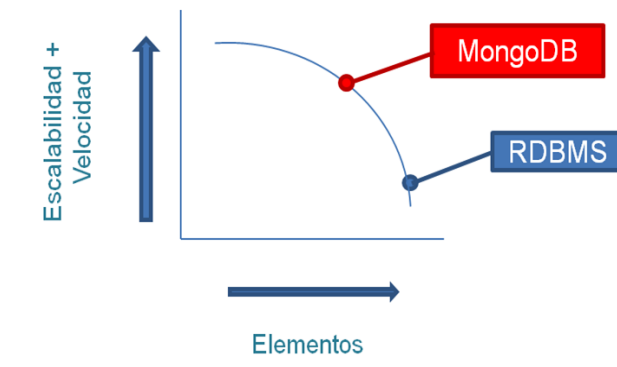
En la siguiente imagen se puede apreciar como MongoDB se encuentra en la zona recomendable, en la que la velocidad y la escalabilidad son altas, así como también es el número de objetos de la base de datos. Por contra en una base de datos relacional (RDBMS) tanto la escalabilidad como la velocidad se ven penalizados ante un número elevado de elementos en la base de datos<sup>22</sup>.

---

<sup>21</sup> J. Fernández (2014), Bases de datos NoSQL III, NoSQL en la práctica. Quantummode.com

<sup>22</sup> Cristian Requena (2010), Que es NoSQL, NoSQL.es. España

Gráfico 2. Imagen tomada de <http://mappinggis.com>



El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software **10gen**. Ahora MongoDB es una base de datos lista para la producción de uso y con muchas características<sup>23</sup>.

- **Escalabilidad.-** La escalabilidad de un sistema informático se detalla como la capacidad de un sistema para expandirse según las necesidades que se obtengan del uso que se le va a dar. Básicamente existen dos formas de escalar.
  - **Escalado vertical.-** Involucra el agregar más recursos al sistema actual para que este logre atender más solicitudes. El escalado vertical mantiene un límite, en el cual llega a un punto en el que la aplicación necesita ser rediseñada para poder escalar más.
  - **Escalado horizontal:** En este proceso el escalado se consigue sencillamente agregando más nodos al sistema. Este tipo de escalado es más fácil y no mantiene conclusiones<sup>24</sup>.

### 2.2.11 Tipos de DBMS de NoSQL

Cuando se comenta algo sobre bases de datos NoSQL, muchas personas se preguntan qué es NoSQL, basta indicar que al día de hoy existen alrededor de 150 sistemas gestores de bases de datos. Elegir uno de ellos puede ser muy difícil, puesto que ninguno ha

<sup>23</sup> Francisco Xavier Ruano V (2014), Análisis y Desarrollo de MongoDB y Redis en Java, España.

<sup>24</sup> Juan Carlos García Candela (2010) Bases de datos NoSQL y escalabilidad horizontal. Universidad de Alicante.

logrado todavía la nombradía que sí bien es cierto han conseguido las bases de datos relacionales.

Pero el principal problema que se sitúa sobre este apartado, es que aunque todas se denominan NoSQL, en realidad hay diferentes tipos. Dependiendo de las necesidades del usuario final, de esta manera se deberá inclinar por el DBMS que más se aproxime a las necesidades propuestas<sup>25</sup>.

De momento existen varias hipótesis diferentes para clasificar las bases de datos NoSQL, el mismo que tenemos el (Teorema CAP, basándonos en el modelo de datos, entre otros), en general se considera que existen cuatro tipos diferentes:

- **Orientadas a documentos.-** Son aquellas BD que tramitan datos semi estructurados. Estos datos son almacenados en algún tipo de formato estándar, el mismo que puede ser XML, JSON o BSON.

Las bases de datos NoSQL son más versátiles que su antecesora SQL. Se pueden emplear en gran cantidad de proyectos, inclusive proyectos que típicamente funcionan sobre bases de datos relacionales<sup>26</sup>.

En esta clase encontramos:

- ✓ **MongoDB:** La BD más famosa del momento. Algunas compañías que actualmente utilizan MongoDB son Foursquare o eBay.
- ✓ **CouchDB:** Esta BD se encuentra orientada a documentos de Apache. Una de las características más relevantes es que los datos son accesibles a través de una API Rest. El mismo que es empleado en corporaciones como Credit Suisse y la BBC.
- **Orientadas a columnas.-** Este tipo de BD fue planteada para efectuar consultas y añadiduras sobre grandes volúmenes de datos.

---

<sup>25</sup> Aurelio Morales (2014), Sobre MongoDB, bases de datos NoSQL y GIS. Formación GIS y difusión tecnológica, España.

<sup>26</sup> José Manuel Alarcón (2014) Fundamentos de bases de datos NoSQL, MongoDB, España.

Funcionan de forma parecida a las bases de datos relacionales, la gran diferencia es que almacenan columnas de datos en lugar de registros.

En esta clase encontramos:

- ✓ **Cassandra:** Sigue un modelo híbrido el mismo que se encuentra entre ser orientada a columnas y clave-valor. Es empleada por Facebook y Twitter.
- ✓ **HBase.** Desarrollada en Java y protegida por el Proyecto Hadoop de Apache, se emplea para procesar grandes volúmenes de datos. Es utilizada por Facebook, Twitter o Yahoo.
- **De clave valor.-** Son las más fáciles de entender. Tan solo guardan secuencias ordenadas que contienen una clave y su valor. Al momento que se desee recuperar un dato, simplemente se busca por su clave y se recupera el valor.

En esta clase encontramos:

- ✓ **DynamoDB:** Desarrollada por Amazon, es una opción de almacenamiento que se puede usar desde los Amazon Web Services. Y es utilizada por Washington Post y Scopely.
- ✓ **Redis:** Desarrollada en C y de código abierto, es manejada por Craigslist y Stack Overflow.
- **En grafo.-** Fundamentadas en la teoría de grafos utilizan nodos para representar los datos almacenados. Son muy útiles para guardar información en modelos con muchas relaciones.

En esta clase encontramos:

- ✓ **Infinite Graph:** Desarrollada en Java y C++ por la compañía Objectivity. Mantiene 2 líneas de licenciamiento: gratuito y de pago.

- ✓ **Neo4j:** Es una BD de código abierto, desarrollada en Java por la compañía Neo Technology. Y utilizada por compañías tales como HP, o Cisco.

Como se puede apreciar las tipologías son muy diferentes.

### **2.2.12 Propiedades para motores de Bases de datos SQL y NoSQL.**

Los DBMS orientados a SQL sigue las propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), por otra parte NoSQL maneja el teorema Brewer o teorema CAP (consistencia, disponibilidad y tolerancia a la partición). A continuación se detalla una breve explicación del funcionamiento de dichas propiedades respectivamente.

- **Propiedades ACID**

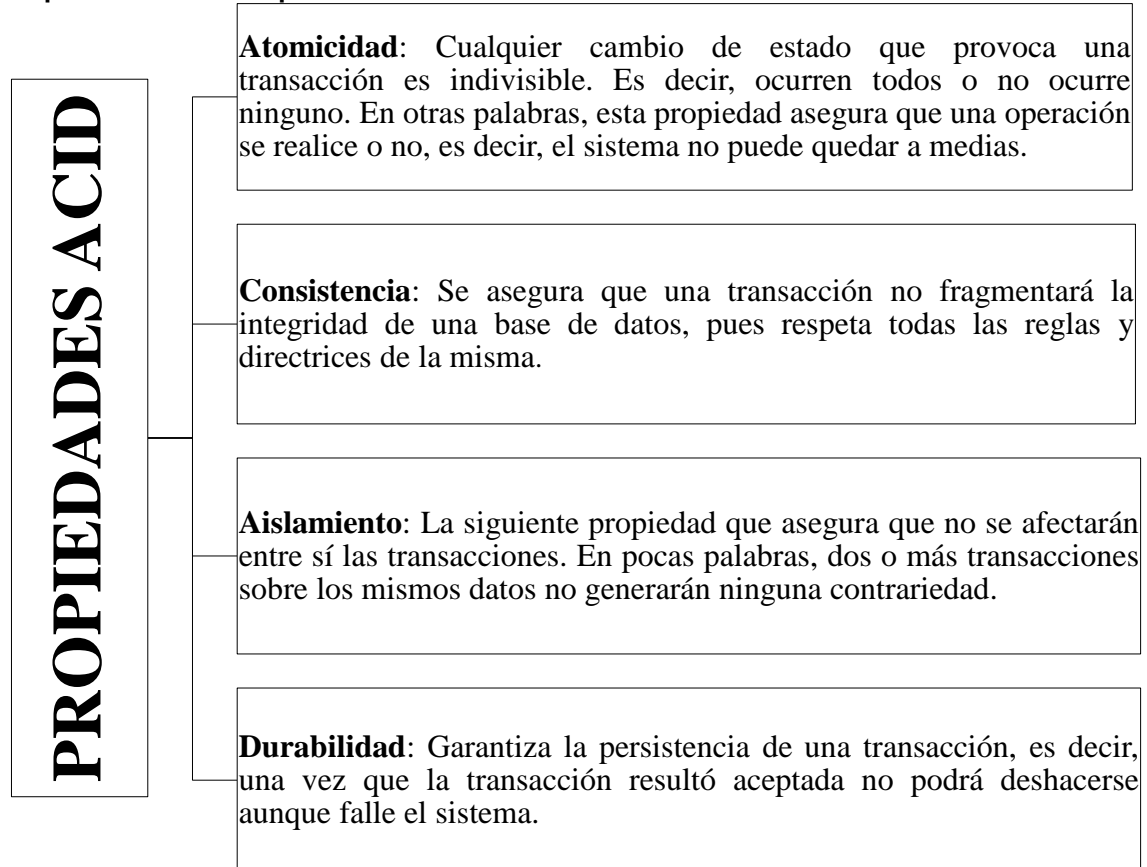
ACID son un conjunto de propiedades necesarias para que las instrucciones, sean consideradas como una transacción en un sistema de gestión de bases de datos. Una transacción es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o elemental<sup>27</sup>.

En 1970, Jim Gray definió las propiedades que necesitaba tener una transacción confiable, y desarrolló tecnologías para automatizarlas. Más tarde, en 1983, Andreas Reuter y Theo Härder crearon el término "ACID" para describir estas 4 propiedades.

---

<sup>27</sup> Leonardo D (2013) ACID en las bases de datos, Argentina

## Propiedades ACID empleadas en motores de base de datos SQL.



Fuente: [www.dosideas.com/](http://www.dosideas.com/) Buenos Aires - Argentina

### • Teorema Brewer o Teorema CAP

El Teorema CAP para sistemas de datos distribuidos, también conocido como "Teorema de Brewer" (El profesor Eric Brewer, de la Universidad de Berkeley, enunció las tres dimensiones de los sistemas distribuidos de almacenamiento de dato, y fue formalmente definida en 2002, convirtiéndola en teorema), la misma que establece "Un sistema de datos compartidos puede asegurar como mucho dos de estas tres propiedades: Consistencia, Disponibilidad y Tolerancia a particiones."<sup>28</sup>

Según el teorema no es posible cumplir con las tres dimensiones a la vez. Hay que elegir dos de ellas, y de las 3 parejas resultantes se generan diferentes tipos de sistemas distribuidos de almacenamiento:

<sup>28</sup> Cecilio Álvarez Caules  
<http://www.cantabriatic.com/>

(2014 /31 /diciembre) NoSQL y el Teorema de CAP,

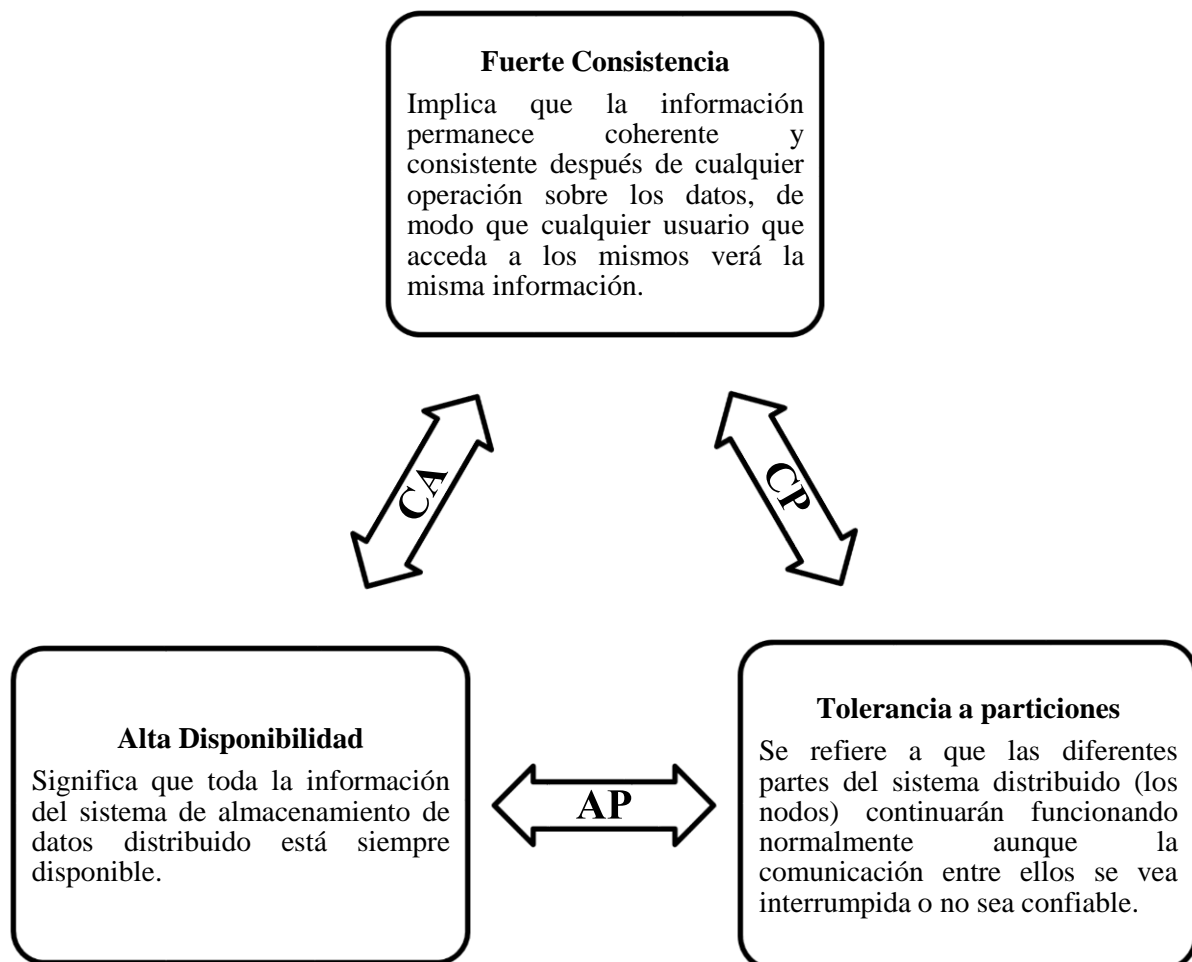
✓ **C y A:** Son métodos en los cuales si falla la comunicación entre sus nodos el conjunto no podrá laborar.

✓ **C y P:** En este apartado si algo sucede una parte de la información no se encontrará utilizable, pero si seguirán funcionando y la información disponible será sólido.

✓ **A y P:** Se presenta durante un fallo de uno de los nodos, la información se encontrará disponible pero puede que no sea sólida.

Por medio de estos pares se puede especificar los sistemas de almacenamiento de datos y saber al dedillo a qué tipología pertenece cada uno de los sistemas más populares, conociendo en cada caso qué características nos pueden proporcionar y adaptar a nuestras necesidades.

**Teorema CAP empleado en motores de base de datos NoSQL.**



Fuente: The Init 2012/04/24

## **CAPÍTULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**



### **3.1. Objeto de estudio realizado**

El detallado estudio del presente proyecto se lo realizaron sobre el rendimiento y estudio comparativo<sup>29</sup> de las semejanzas, diferencias y mejoras existentes entre SQL y NoSQL.

### **3.2. Tipo de Investigación.**

#### **3.2.1. Exploratoria**

Para explorar un tema comparativamente desconocido como lo es NoSQL, se dispone de un amplio conjunto de medios y técnicas para recolectar datos en diferentes ciencias, la investigación exploratoria terminará cuando haya sido posible crear un medio lo suficientemente consistente como para determinar que indicadores son relevantes al problema a partir de los datos recolectados y por lo tanto deben ser investigados.

#### **3.2.2. Bibliográfica**

- Se la obtuvo en la recopilación de los datos, valiéndose del manejo adecuado de libros, internet, artículos y documentos relacionados al presente Proyecto de Investigación.

### **3.3. Método de investigación.**

#### **3.3.1. Deductivo**

Con este método se podrá determinar que gestor se va a utilizar con la información que consta en la base de datos para las pruebas.

#### **3.3.2. Analítico**

Empleado para analizar la relación existente entre los gestores de datos SQL y NoSQL, para ofrecer mayor seguridad y rendimiento.

#### **3.3.3. Sintético**

Sirve para sintetizar y emitir conclusiones acerca de la aplicabilidad de cada uno de los gestores de datos.

---

<sup>29</sup> Francisco Montiel Vera (2006), Análisis y estudio comparativo de los programas de computación para la administración de proyectos, Tesis de Ingeniero Industrial. Universidad Autónoma del Estado de Hidalgo, Pachuca.

### 3.4. Fuentes de recopilación de información.

#### 3.4.1. Fuentes Primarias.

- ✓ Sondeos
- ✓ Prácticas de factibilidad
- ✓ Usos en la Bases de datos reales.

#### 3.4.2. Fuentes Secundarias.

- ✓ Revistas
- ✓ Ensayos
- ✓ Documentos
- ✓ Internet

**Tabla 4. Materiales y recursos empleados para el desarrollo de la investigación.**

Descripción	Cantidad
✓ Fuentes de investigación virtuales	37
✓ Internet	500 hrs/h
✓ Computadores	2
✓ Impresora	3
✓ Remas de papel	6
✓ Libros	9
<b>Recursos.</b>	
Descripción	Cantidad
✓ Autor del tema de investigación	1
✓ Tutor encargado del tema de investigación	1
✓ Biblioteca	1
✓ Windows 7, 8	-
✓ Microsoft Word	-
✓ Microsoft Excel	-
✓ MySql Server	-
✓ MongoDB	-

**Tabla 5. Presupuesto del Proyecto de Investigación.**

DESCRIPCIÓN	CANTIDAD	V. UNITARIO	TOTAL
COMPUTADORA	1	1,800	1,800
HORAS INTERNET	500 Hrs	0,21	105,00
HORAS HOMBRE	800 S	8,00	6,400
RESMA DE PAPEL	7	6,00	42,00
CARTUCHOS DE TINTA	3	15,00	45,00
BOLÍGRAFOS	4	0,75	3,00
PENDRIVE	1	10,00	10,00
CDS	7	2,00	14,00
ANILLADOS	7	1,50	10,50
OTROS	-	50,00	50,00
TOTAL	-	-	8479,50

**CAPÍTULO IV.**

**RESULTADOS Y DISCUSIÓN**

## 4.1. ESPECIFICACIÓN DE LOS DATOS UTILIZADOS

Para la elaboración de la investigación se obtuvo información de una base de datos llamada “Centro\_Comercial”, la misma que mantiene 69255 registros (sesenta y nueve mil doscientos cincuenta y cinco) y 128313 usuarios (siento veintiocho mil trecientos trece), mantiene relación de correspondencia, además mantiene un total de 24 tablas, 15 procedimientos almacenados, a continuación se describe el nombre de las tablas.

3. Cantón	14. Seguimiento_sistema
4. Categoría	15. País
5. Clientes	16. Pedidos
6. Compras	17. Periodos
7. Empleados	18. Proveedor
8. Factura	19. Provincia
9. Fecha	20. Producto
10. Tipo_proveedor	21. Proformas
11. Tipo_producto	22. Usuario
12. Tipo_empleado	23. Ventas
13. Tipo_cliente	

### 4.1.1. Modelo Relacional SQL

Para el tratamiento de los datos en las pruebas a realizar, se seleccionó por parte de SQL uno de los tres DBMS más utilizados, **MySQL** mismo que fue escogido por sus características de similitud con MongoDB, por su preferencia y facilidad de uso<sup>30</sup>.

Para colocar la base de datos en MySQL, se importaron los datos utilizando el nombre del usuario al que se desea realizar la afectación, el nombre de la base de datos, y el comando **SQL.data**.

---

<sup>30</sup> (Silberschatz, Korth, & Sudarshan, 1998), S. (1998). Fundamentos de bases de datos. (Tercera edición). Madrid: McGraw-Hill.

#### 4.1.2. Modelo No Relacional NoSQL

Para el desarrollo de las pruebas se utilizó por parte de NoSQL, el gestor de bases de datos **MongoDB**, mismo que fue elegido por sus características, evolución, preferencia a la hora de escoger un gestor y por su uso comprensible.

Para colocar la base de datos en MongoDB, se puede utilizar la herramienta **MongoHub**, que tiene una utilidad nativa para importar bases de datos desde MySQL, o simplemente podríamos exportar los registros de MySQL a un archivo de texto plano e importarlos directamente a la consola de MongoDB utilizando el comando **Mongoimport**.

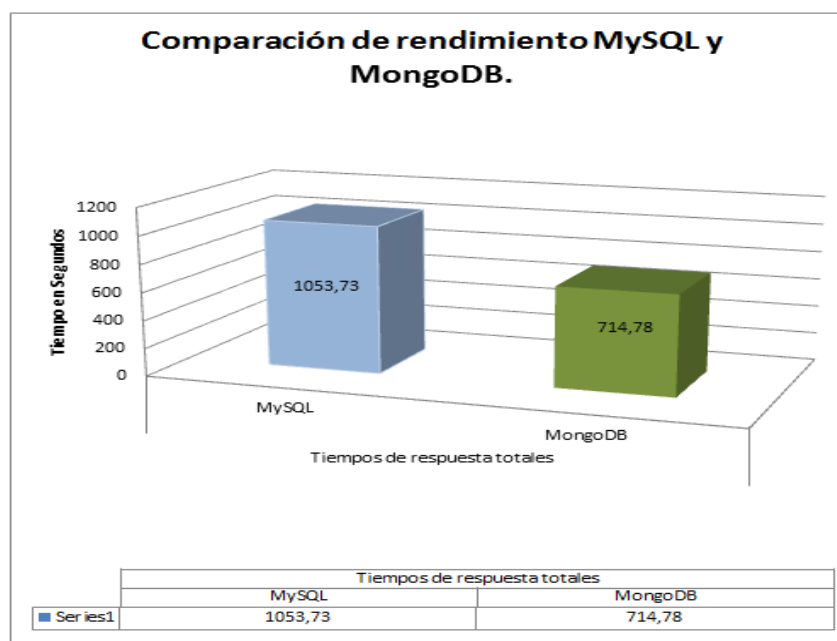
### 4.2. RENDIMIENTO DE LOS GESTORES DE BASES DE DATOS.

El rendimiento es uno de los indicadores que se debe tener en cuenta al momento de seleccionar el gestor de bases de datos que más se ajuste a las necesidades del usuario, es por este motivo que se determinaron los elementos a tener en cuenta mediante la solución de un ejercicio utilizando una base de datos para realizar las pruebas.

El soporte principal del rendimiento se basa en mantener la integridad de los datos, el problema con las RDBMS, radica en que cada vez que se debe organizar la información resulta ser un proceso muy costoso, directamente proporcional al volumen de datos, mientras más grande, más tiempo y recursos serán necesarios para que lograr ordenar y procesar los información, es por eso que no se normaliza y se permite la redundancia de los datos.

Ahora por ejemplo: Si se tiene una empresa con información distribuida en varios servidores de bases de datos, sin importar del tipo que sea, necesitamos procesar todos estos volúmenes de datos, su funcionamiento será el de diversos servidores buscando información al mismo tiempo, ordenándola y procesándola de manera que permita presentar un par de números en cuestión de segundos, esto es lo que hacen las bd NoSQL.

**Gráfico 3. Representación gráfica de la comparación realizada entre gestores de base de datos SQL y NoSQL**



Fuente: Moreira, W. (2016)

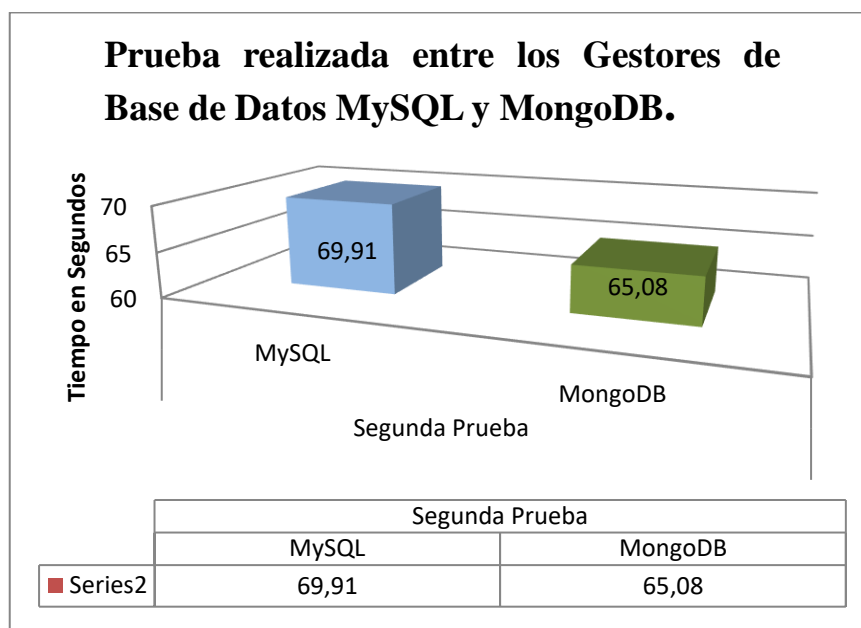
#### **4.3. ANALISIS DE SEGURIDAD DE LOS MOTORES DE BASE DE DATOS.**

Otro de los indicadores a tener en cuenta al momento de seleccionar el gestor de bases de datos que más se ajuste a las necesidades del usuario es la seguridad, ya que de esto depende la autenticación, confidencialidad, vulnerabilidades, e integridad de los datos.

SQL y las NoSQL son a menudo empleadas para sucesos muy diferentes, de hecho se debe conocer qué tipo de base de datos utilizar dependiendo del proyecto, en la actualidad es opcional, es decir, cuando se necesite alta seguridad de datos, explotación de datos en estadísticas, resúmenes, procesos con SP o triggers, es allí cuando se debe escoger cual es el motor que mejor se ajusta a las necesidades del usuario.

Además de que MySQL mantiene varias opciones, además del hecho que es un software libre (gratuito) mientras que MongoDB no contiene demasiadas medidas de seguridad.

**Gráfico 4. Representación gráfica del nivel de seguridad obtenido al utilizar el autenticador de usuarios de SQL y NoSQL**



Fuente: Moreira, W. (2016)

#### **4.4. VERIFICAR LA INTEGRIDAD DE LOS DATOS AL SER EMPLEADOS EN SQL Y NOSQL.**

Siguiendo una filosofía donde los indicadores tales como el rendimiento y la seguridad son primordiales se prioriza también la integridad de los datos. Por ello es necesario utilizar frecuentemente algún tipo de mecanismo complementario (Auditorías informáticas), ajeno al motor de base de datos para asegurar su integridad.

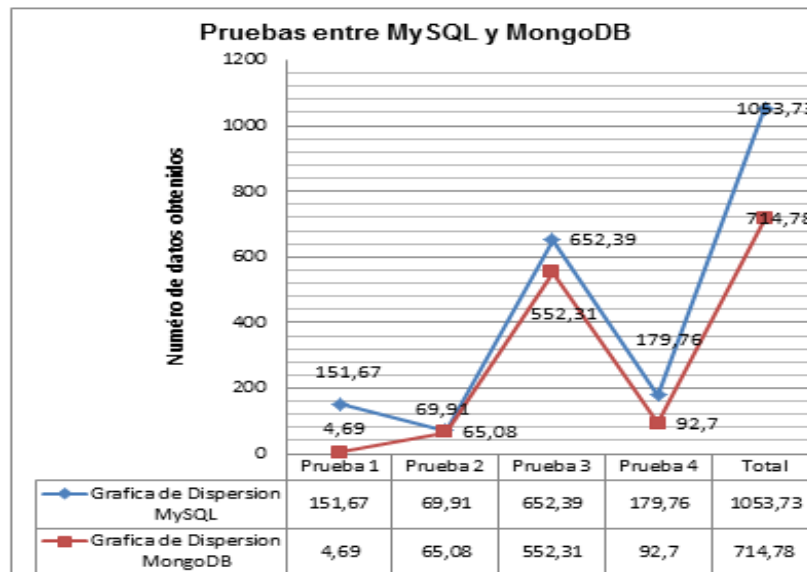
Las bases de datos SQL, han servido mucho para hacer aplicaciones transaccionales en las que mantener y proteger la integridad de la información es vital; mientras que por otra parte tenemos las NoSQL, que abogan más por la evolución del Internet, misma que radica en los nuevos desarrollos, cabe indicar que con NoSQL no se garantiza la integridad de los datos, pero existen métodos para poder hacer esto con bases de datos documentales.

Las bases de datos relacionales hacen uso de un lenguaje de consulta estructurado llamado SQL, la idea era organizar la información en conjuntos de datos, y así poder mantener una coherencia entre ellos (integridad), fue creciendo el volumen de información pero pocos tenían acceso a manipularla, mientras la



Internet fue expandiéndose, cada vez más personas acceden a los datos, para ello los RDBMS son muy lentos y de la forma en la que fueron diseñadas traerían problemas, aquí actúan las NoSQL, una forma de almacenar y manipular los datos sin necesidad de ser restrictivo como el SQL, con un objetivo muy básico, sacrificar integridad por velocidad.

**Gráfico 5. Representación gráfica de varias pruebas realizadas entre SQL y NoSQL para determinar el grado de integridad que mantienen los datos**



Fuente: Moreira, W. (2016)

## APLICACIÓN DEL ESTUDIO COMPARATIVO ENTRE SQL Y NOSQL

A continuación se detallan los resultados obtenidos de la presente investigación y las condiciones con las que se realizaron las pruebas de rendimiento, las mismas que detallaremos en el siguiente apartado:

- Exponer los indicadores más relevantes a la hora de escoger un determinado gestor de base de datos empleado en grandes volúmenes de datos.
- Establecer textualmente las características más relevantes observadas entre SQL y NoSQL.

- Demostrar los resultados obtenidos de las pruebas en base a rendimiento, seguridad e integridad de datos, realizadas entre ambos gestores de base de datos, e indicar una conclusión veraz con base a los indicadores antes establecidos.
- A la hora de escoger una base de datos se debe de tomar en cuenta principalmente la disponibilidad, complejidad, y tiempos de respuesta.
- Se logró realizar un estudio comparativo realizado entre SQL y NoSQL por medio del cual obtener los tiempos de respuesta, mismos empleados para determinar la viabilidad de cada uno de ellos.

### **Pruebas realizadas entre una base de datos MySQL y MongoDB.**

Después de realizar varios estudios en cuanto a velocidad equivalente a bases de datos SQL y NoSQL, probaré la velocidad de cada uno de estos motores. Para ello se han preparado 4 test de prueba para demostrar y garantizar cual es el más recomendable con base a rendimiento, seguridad e integridad de datos.

### **Procedimientos realizados para la obtención de resultados.**

Antes de describir el proceso de la instalación es preciso indicar que las mismas fueron realizadas en un sistema operativo Windows 7 de 64 bits. Ya que al utilizar el motor de bases de datos “MongoDB” en un sistema operativo de 32 bits se tiene una capacidad de almacenamiento de bases de datos de un máximo de hasta 2 Gb.

A continuación se especifica como iniciar una base de datos en MySQL y en MongoDB.

Gráfico 6. Consola de Inicio en MySQL

```
mysql> select count(*) from tweets;
+-----+
| count(*) |
+-----+
|    69255 |
+-----+
1 row in set (0.01 sec)

mysql> select count(*) from users;
+-----+
| count(*) |
+-----+
|    12313 |
+-----+
1 row in set (0.00 sec)
```

Gráfico 7. Consola de Inicio en la consola MongoDB

```
> db.tweets.count()
69255
> db.users.count()
12313
```

#### 4.5. Primer prueba de rendimiento

Realizar un count de los tweets 10 000 (diez mil) veces.

Gráfico 8. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL

```
cuxibamba:mysql macool$ ruby count.rb
Rehearsal -----
count 10000 times  6.250000  0.330000  6.580000 (151.418105)
----- total: 6.580000sec

               user      system      total      real
count 10000 times  6.290000  0.320000  6.610000 (151.667663)
```

**Gráfico 9. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB**

```
cuxibamba:mongodb macool$ ruby count.rb
Rehearsal -----
count 10000 times    3.810000    0.300000    4.110000 ( 4.685105)
----- total: 4.110000sec

              user      system      total      real
count 10000 times    3.820000    0.290000    4.110000 ( 4.685353)
```

**Resultados obtenidos:**

- **MySQL:** 151.67 segundos / 2,53 min
- **MongoDB:** 4.69 segundos / 0,08 min

#### 4.6. Segunda prueba de rendimiento

“Consultar una palabra aleatoria 500 veces (where text like '%word%'). De la siguiente manera:

- Contar cuántas veces aparece dentro del contenido de los tweets
- Hacer la proyección de los resultados; no solamente el conteo

**Gráfico 10. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL**

```
cuxibamba:mysql macool$ ruby query.rb
Rehearsal -----
count query for random word 500 times    0.440000    0.020000    0.460000 ( 28.596938)
select query for random word 500 times    0.730000    0.030000    0.760000 ( 74.751770)
----- total: 1.220000sec

              user      system      total      real
count query for random word 500 times    0.350000    0.010000    0.360000 ( 26.407030)
select query for random word 500 times    0.490000    0.020000    0.510000 ( 69.910461)
```

**Gráfico 11. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB**

```
cuxibamba:mongodb macool$ ruby query.rb
Rehearsal -----
count query for random word 500 times    0.350000    0.030000    0.380000 ( 69.295281)
select query for random word 500 times    0.410000    0.010000    0.420000 ( 68.140760)
----- total: 0.800000sec

              user      system      total      real
count query for random word 500 times    0.340000    0.020000    0.360000 ( 63.709326)
select query for random word 500 times    0.320000    0.020000    0.340000 ( 65.082841)
```

#### Resultados obtenidos:

- **MySQL:** 69.91 segundos al hacer la proyección y asignar los resultados a variables. / 1,17 min
- **MongoDB:** 65.08 segundos al hacer la proyección y asignar los resultados a variables. / 1,08 min

#### 4.7. Tercera prueba de rendimiento

“Hacer un Select que busque por id (clave primaria) 1 000 000 (un millón) de veces.

**Gráfico 12. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL**

```
cuxibamba:mysql macool$ ruby find.rb
Rehearsal -----
finding rand tweet 1000000 times 557.280000    21.470000    578.750000 (677.584838)
----- total: 578.750000sec

              user      system      total      real
finding rand tweet 1000000 times 536.040000    20.640000    556.680000 (652.392665)
```

**Gráfico 13. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB**

```
cuxibamba:mongodb macool$ ruby find.rb
Rehearsal -----
finding rand tweet 1000000 times 460.530000    27.210000    487.740000 (548.675625)
----- total: 487.740000sec

              user      system      total      real
finding rand tweet 1000000 times 464.000000    27.400000    491.400000 (552.508299)
```

## Resultados obtenidos:

- **MySQL:** 652.39 segundos / 10,87 min
- **MongoDB:** 552.51 segundos / 9,21 min

### 4.8. Cuarta prueba de rendimiento

Realizar un ejercicio utilizando el CRUD, compuesto por lo siguiente:

- Crear 10000 (diez mil) tweets.
- Encontrarlos (por su id) y actualizar su texto.
- Buscar nuevamente los 10000 tweets y actualizar tres veces el contenido de cada uno, guardando en DB cada vez que se actualice.
- Encontrar cada uno de los 10000 tweets (por su id) y eliminarlo.

**Gráfico 14. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MySQL**

```
cuxibamba:mysql macool$ ruby crud.rb
Rehearsal -----
creating 10000 tweets      39.190000    0.800000    39.990000 ( 44.653293)
find and update them      21.950000    0.960000    22.910000 ( 28.654262)
updating each one three times 64.350000    2.740000    67.090000 ( 81.312197)
removing them             13.400000    0.810000    14.210000 ( 18.313242)
----- total: 144.200000sec

              user      system      total      real
creating 10000 tweets      35.330000    0.730000    36.060000 ( 40.181477)
find and update them      23.100000    0.960000    24.060000 ( 29.710288)
updating each one three times 69.420000    2.920000    72.340000 ( 87.323573)
removing them              17.130000    0.930000    18.060000 ( 22.551120)
```

**Gráfico 15. Pantalla con los detalles obtenidos de la consulta realizada en la consola de MongoDB**

```
cuxibamba:mongodb macool$ ruby crud.rb
Rehearsal -----
creating 10000 tweets      5.050000    0.170000    5.220000 ( 5.212017)
find and update them      11.020000    0.520000    11.540000 ( 12.733068)
updating each one three times 34.080000    1.580000    35.660000 ( 39.534616)
removing them              8.170000    0.500000    8.670000 ( 9.995912)
----- total: 61.090000sec

              user      system      total      real
creating 10000 tweets      4.650000    0.160000    4.810000 ( 4.806377)
find and update them      15.530000    0.800000    16.330000 ( 18.129270)
updating each one three times 46.710000    2.410000    49.120000 ( 54.862238)
removing them              12.240000    0.760000    13.000000 ( 14.901368)
```

## Resultados obtenidos:

### MySQL:

- 40.18 segundos para almacenar diez mil tweets
- 29.71 segundos para encontrar y actualizar diez mil tweets

- 87.32 segundos para encontrar los tweets y actualizar tres veces cada uno
- 22.55 segundos para eliminar los diez mil tweets

Total: 179.76 segundos / 3 minutos

#### **MongoDB:**

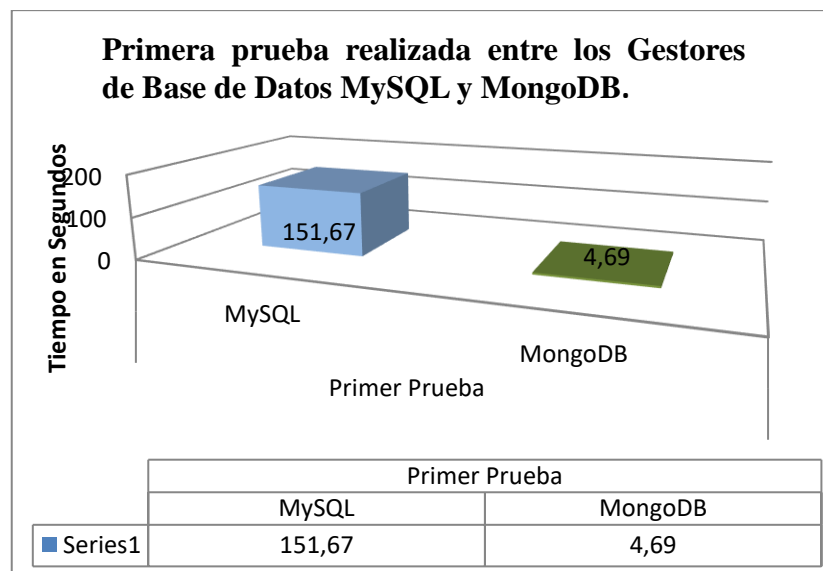
- 4.81 segundos para almacenar diez mil tweets
- 18.13 segundos para encontrar y actualizar diez mil tweets
- 54.86 segundos para encontrar los tweets y actualizar tres veces cada uno
- 14,9 segundos para eliminar diez mil tweets

Total: 92.7 segundos / 1,55 minutos (casi la mitad de MySQL)

#### **4.9. Resultados Gráficos de la Investigación.**

Para comprobar gráficamente las diferencias de NoSQL utilizando el motor de base de datos MongoDB y SQL que aplica el DBMS MySQL, a continuación se detallan las pruebas presentadas mediante gráficos estadísticos, los cuales nos muestran una amplia ventaja en cuanto a rendimiento de MongoDB.

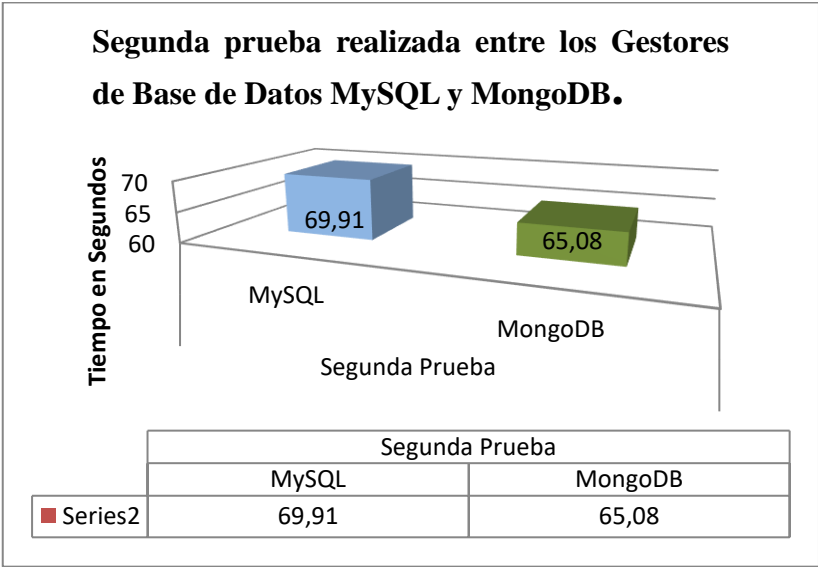
**Gráfico 16. Representación gráfica de la primera prueba realizada entre SQL y NoSQL**



Esta pregunta se efectuó con el propósito de conocer qué gestor de bases de datos es el más relevante, teniendo en cuenta los indicadores necesarios para obtener resultados válidos.

Los resultados obtenidos por esta gráfica indican que MongoDB en comparación a la consola de MySQL, ha logrado obtener tiempos de respuesta de 2,33 minutos para MySQL, mientras que para MongoDB tan solo 0,08 minutos.

**Gráfico 17. Representación gráfica de la segunda prueba realizada entre SQL y NoSQL**

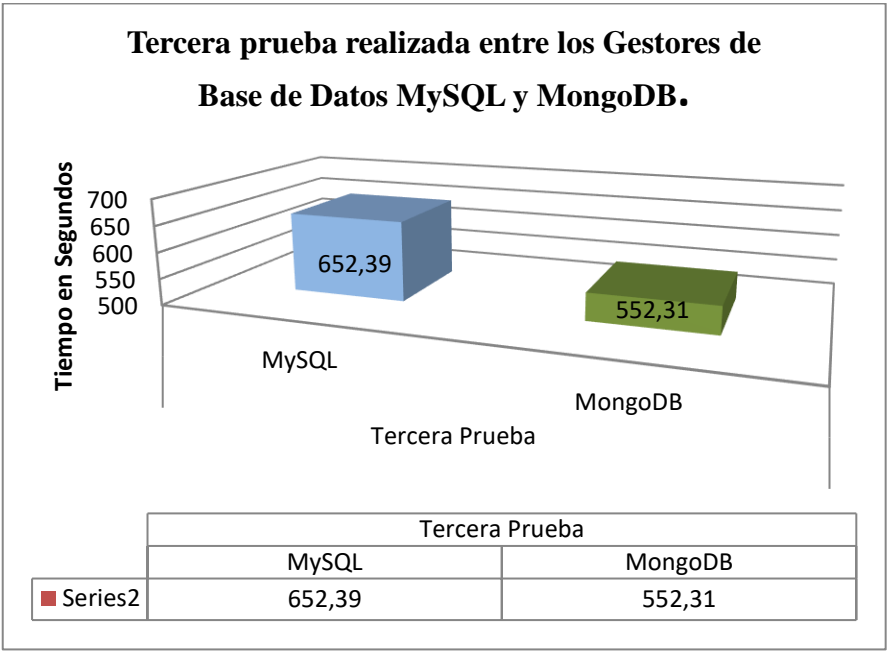


Los datos obtenidos en esta gráfica se utilizaron para determinar que indicadores influyen en él, para ello se empleó una consulta un tanto más estructurada que la anterior.

En esta gráfica se puede observar los gestores de bases de datos, obteniendo resultados no esperados ya que tanto MongoDB como MySQL, logran obtener tiempos de respuesta semejantes tales como son 1,17 minutos para MySQL, mientras que para MongoDB 1,08 minutos.

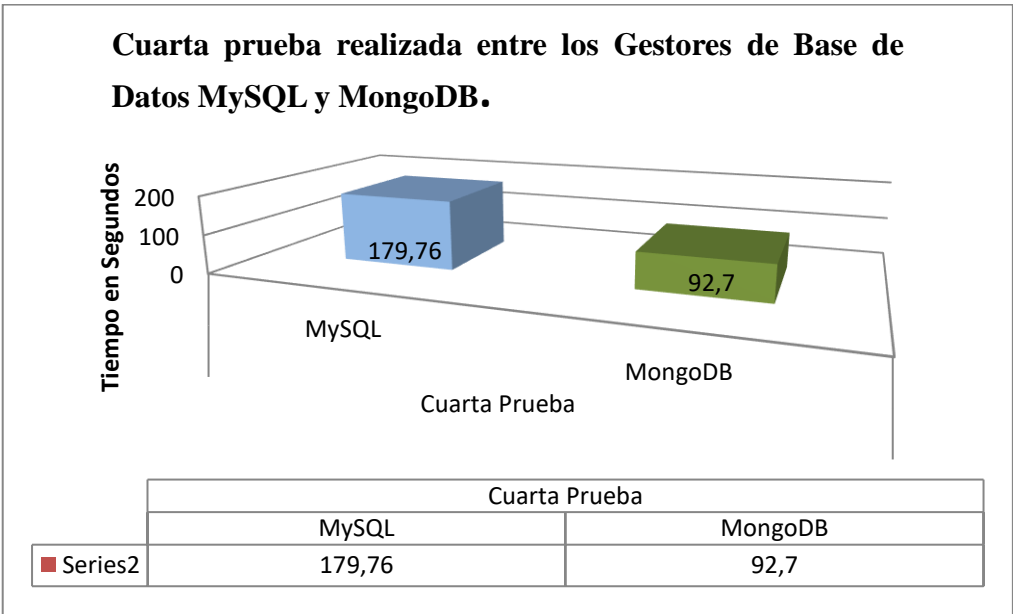


**Gráfico 18. Representación gráfica de la Tercera prueba realizada entre SQL y NoSQL**



La gráfica indica que se obtuvo resultados poco aceptables ya que tanto MongoDB como MySQL, logrando obtener tiempos de respuesta un tanto similares, tales como son 10,87 minutos para MySQL, mientras que para MongoDB 9,21 minutos.

**Gráfico 19. Representación gráfica de la Cuarta prueba realizada entre SQL y NoSQL**



Gracias a la representación gráfica de la consulta realizada para obtener tiempos de respuesta entre los gestores de bases de datos, empleando CRUD<sup>31</sup> se alcanzó percibir la diferencia entre ambos Motores, ya que al realizar las 4 pruebas realizadas, Crear, Obtener, Actualizar y Borrar, MySQL obtuvo un total de 3 minutos, mientras que MongoDB realizó el mismo proceso en tan solo 1,55 minutos.

Una vez obtenidos los datos necesarios al realizar pruebas entre bases de datos relacionales y no relacionales, y después de haber realizado una profunda investigación, se puede ultimar que en aspecto de rendimiento los gestores NoSQL son superiores, sin desmerecer a los motores SQL que efectúan las pruebas en un tiempo menor pero manteniendo los datos íntegramente, algo que generalmente Not Only SQL descarta para ejecutar en menor tiempo las consultas.

Según el portal tecnológico DB – ENGINES dentro de las bases de datos más usadas encontramos tres del tipo no relacional, las mismas que son: **MongoDB, Cassandra y Redis**, ambas están incrementando su adopción a nivel global, según el portal.

A continuación se muestra el top de los DBMS más usados registrados con fecha de actualización Octubre del 2015 en esta lista que alberga a un total de 283 sistemas de gestión de bases de datos.

---

<sup>31</sup> Nick Salloum (2015) CRUD (Create, Read, Update, Delete)

**Gráfico 20. En la gráfica que se presenta a continuación se puede apreciar los 10 DBMS más usados hasta Octubre del 2015.**

283 systems in ranking, October 2015

Rank			DBMS	Database Model	Score		
Oct 2015	Sep 2015	Oct 2014			Oct 2015	Sep 2015	Oct 2014
1.	1.	1.	Oracle	Relational DBMS	1466.95	+3.58	-4.95
2.	2.	2.	MySQL	Relational DBMS	1278.96	+1.21	+15.99
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1123.23	+25.40	-96.37
4.	4.	↑ 5.	MongoDB +	Document store	293.27	-7.30	+52.86
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	282.13	-4.05	+24.41
6.	6.	6.	DB2	Relational DBMS	206.81	-2.33	-0.86
7.	7.	7.	Microsoft Access	Relational DBMS	141.83	-4.17	+0.19
8.	8.	↑ 10.	Cassandra +	Wide column store	129.01	+1.41	+43.30
9.	9.	↓ 8.	SQLite	Relational DBMS	102.67	-4.99	+7.71
10.	10.	↑ 12.	Redis +	Key-value store	98.80	-1.86	+19.42

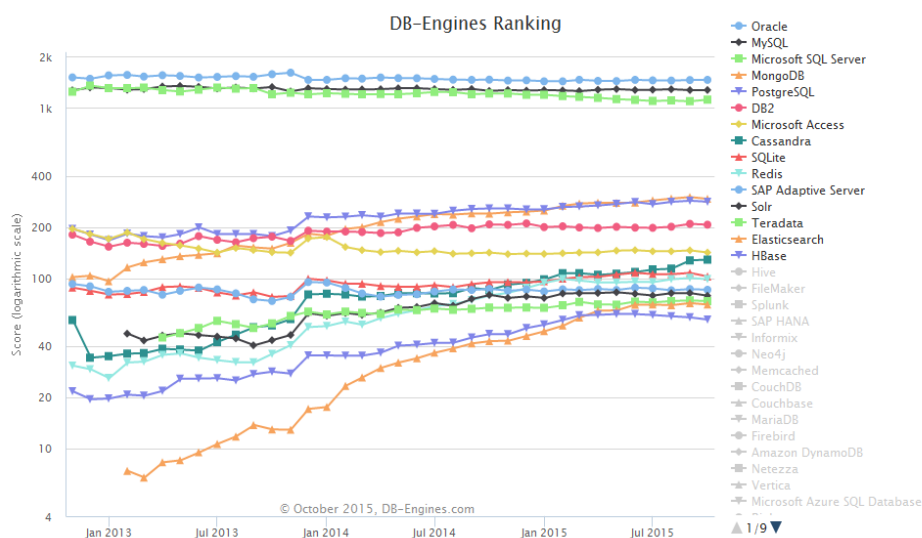
Las bases de datos no relacionales vienen ganando terreno ampliamente, es el caso de MongoDB, Cassandra y Redis que se puede observar en la gráfica que han escalado sobre SQLite e inclusive sobre PostgreSQL, datos actualizados a Octubre del 2015, y que en la siguiente gráfica se puede observar cómo se han mantenido hasta la actualidad.

**Gráfico 21. En la gráfica que se presenta a continuación se puede apreciar los 10 DBMS más usados hasta febrero del 2016.**

296 systems in ranking, February 2016

Rank			DBMS	Database Model	Score		
Feb 2016	Jan 2016	Feb 2015			Feb 2016	Jan 2016	Feb 2015
1.	1.	1.	Oracle	Relational DBMS	1476.14	-19.94	+36.42
2.	2.	2.	MySQL	Relational DBMS	1321.13	+21.87	+48.67
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1150.23	+6.16	-27.26
4.	4.	4.	MongoDB +	Document store	305.60	-0.43	+38.36
5.	5.	5.	PostgreSQL	Relational DBMS	288.66	+6.26	+26.32
6.	6.	6.	DB2	Relational DBMS	194.48	-1.89	-7.94
7.	7.	7.	Microsoft Access	Relational DBMS	133.08	-0.96	-7.47
8.	8.	8.	Cassandra +	Wide column store	131.76	+0.81	+24.68
9.	9.	9.	SQLite	Relational DBMS	106.78	+3.04	+7.22
10.	10.	10.	Redis +	Key-value store	102.07	+0.92	+2.86

## Gráfico 22. Evolución de los gestores de base de datos



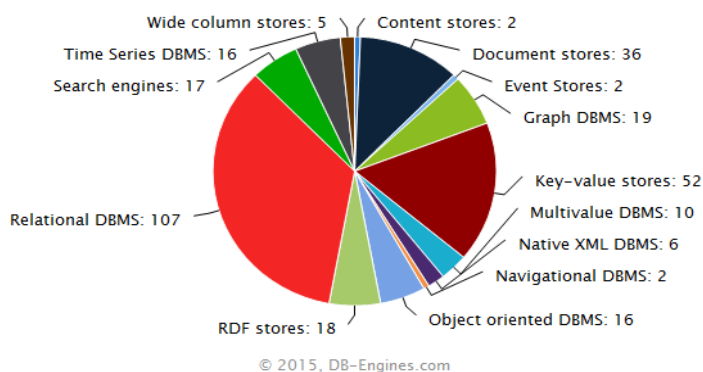
En los últimos años tanto las bases de datos relacionales como las no relacionales han obtenido un crecimiento constante, tal como se aprecia en la gráfica, MongoDB, Cassandra y DB2 para las NoSQL han obtenido una evolución significativa, mientras que Oracle, MySQL y Microsoft SQL Server lo hacen para las bases de datos SQL

## Numero de DBMS relacionales y no relacionales por categoría

Los DBMS se clasifican de acuerdo a su modelo de base de datos (por ejemplo, DBMS relacional, y no relacional (key-value stores, document stores, entre otros.). Este gráfico circular muestra el número de sistemas en cada categoría.

## Gráfico 23. Cantidad de motores de base de datos por categoría.

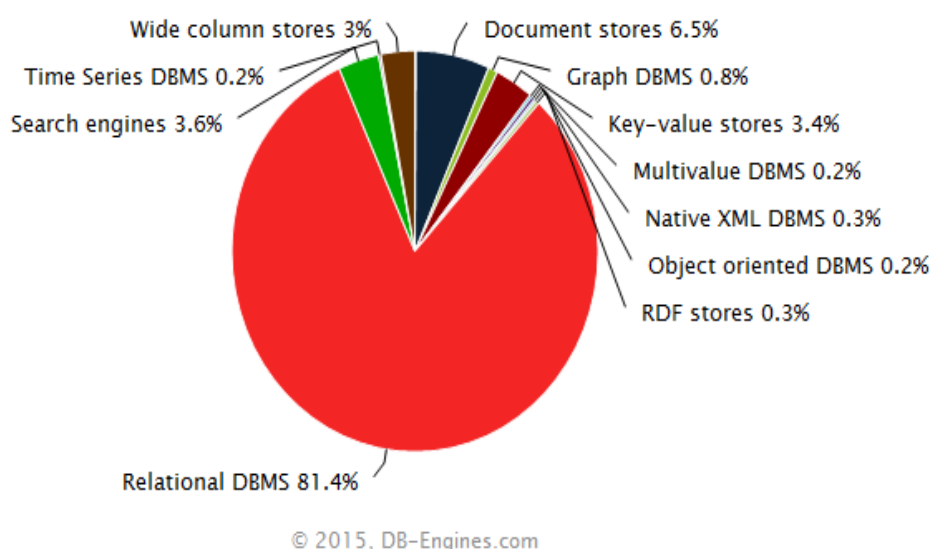
Number of systems per category, October 2015



## Clasificación puntajes por categoría, octubre 2015

Esta gráfica muestra la popularidad de cada categoría. Se calcula con los puntajes obtenidos en el ranking de todos los sistemas individuales por categoría. La suma de todas las puntuaciones de clasificación es de 100%.

**Gráfico 24. Popularidad de los motores de bases de datos**



## Los cambios más populares por categoría

A continuación podremos apreciar la línea de tiempo sobre el crecimiento o deceso de algunos sistemas. Para cada mes los tres mejores sistemas por categoría son elegidos y se calcula la media de sus puntuaciones de clasificación. La mayoría de los sistemas de la tendencia establecida comienza a enero de 2013, pero los motores de búsqueda y DBMS de varios valores solamente se recogen desde febrero de 2013 y mayo de 2013.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

## 5.1 CONCLUSIONES

- Se determinó que al realizar una consulta en una base de datos no relacional, los tiempos de respuesta disminuyen considerablemente en comparación a la misma consulta realizada en una del tipo relacional. No obstante cabe indicar que los tiempos de respuesta para realizar una importación a SQL, es menor al obtenido con una NoSQL.
- La gran cantidad de motores de bases de datos NoSQL, hace que sean una alternativa viable en cuanto a escalabilidad, y rendimiento.
- Una de las principales características al utilizar bases de datos no relacionales es la facilidad de almacenar todo tipo de datos, es decir, si se requiere de un campo extra ya no es necesario rediseñar la estructura de nuestra base de datos.
- Se consiguió identificar funcionalidades tales como la capacidad de procesar grandes cantidades de información, en donde se puede diseñar diferentes aplicaciones dependiendo de los requerimientos del usuario, y sobre todo que son adaptables a tecnologías existentes.
- Se logró justificar que a pesar de que el tamaño de las bases de datos no relacionales sea mucho más grande que el de una base de datos relacional los tiempos de ejecución son más eficientes, no obstante, tiende a perder integridad de los datos por ganar velocidad.
- Además NoSQL es una alternativa a los sistemas tradicionales basados en SQL, ya que la información se almacena como tablas con columnas dinámicas, es decir pueden existir requerimientos con más datos que otras, haciendo más sencillos los cambios de la estructura de la información sin necesidad de rediseñar el procedimiento.

## 5.2 RECOMENDACIONES

A continuación se mencionan algunos aspectos a tener en cuenta para el desarrollo de proyectos con gestores de bases de datos relacionales o no relacionales:

- Al desarrollar bases de datos utilizando MongoDB, se deberá tener en cuenta las propiedades, características, ventajas y desventajas que brinda un sistema operativo ya sea de 32 bits o de 64 bits; debido a que un SO de 32 bits administra BD de máximo hasta 2 Gb.
- Los motores de bases de datos relacionales, son una buena alternativa para mantener un buen diseño de la base de datos, y evitar la pérdida de información en grandes cantidades.
- Se debe tener en cuenta la utilización del tipo de software a emplear, ya sea de licencia libre o pagada, para disminuir costos y mantener una correcta administración de la información.
- Las BD NoSQL son muy buenas al momento de trabajar con una gran cantidad de información; pero las SQL mantienen ventajas gracias a que la pérdida de la información es mínima, siendo un punto vital para ciertas empresas.



## **CAPÍTULO VI**

### **BIBLIOGRAFÍA**

## 6.1. LITERATURA CITADA

- 1) P.D.I.I Francisco Montiel Vera (2006), Análisis y estudio comparativo de los programas de computación para la administración de proyectos, Tesis de Ingeniero Industrial. Universidad Autónoma del Estado de Hidalgo, Pachuca.
- 2) Carmen (Marín, 2002)Marín Escofet (2002), El lenguaje SQL. Universidad Virtual UOC (Universidad Oberta de Catalunya), España.
- 3) Groff, J.R.; Weinberg, P.N. (1998). LAN Times. Guía de SQL. Osborne: McGraw-Hill.
- 4) (Silberschatz, Korth, & Sudarshan, 1998), S. (1998). Fundamentos de bases de datos. (Tercera edición). Madrid: McGraw-Hill.
- 5) Melton, J.; Simon, A.R. (2001). SQL (1999). Undestandign Relational Language Components. Morgan Kaufmann.
- 6) Javier Garzás (2013), ¿Bases de datos NoSQL o Bases de datos SQL? ¿Tiene sentido en “nuestro mundo” usar bases de datos NoSQL?. JavierGarzas.com
- 7) Ana Milena Barragan Charry, Andrea Forero Sanabria (2013), implementación de una base de datos NoSQL para la Generación de la matriz o/d. Universidad Católica de Colombia, Bogotá. (Barragan, 2013)
- 8) Ing. Esteban Zapata Rojas (2013), SQL vs. NoSQL ¿Cuál es el mejor?. estebanz01.wordpress.com
- 9) Aurelio López Ovando (2012), Almacenamiento de base de datos. aureliux.files.wordpress.com.
- 10)Juan Diego Lopera Echavarría, Carlos Arturo Ramírez Gómez, Marda Ucaris Zuluaga Aristizábal, Jennifer Ortiz Vanegas (2010), El método analítico como método natural. Universidad de Antioquia, Colombia.

- 11)** Margaret Rouse (2015), No solo SQL (NoSQL). Barney Beal. Searchdatacenter.com
- 12)** J. Fernández (2014), Bases de datos NoSQL I, Que es NoSQL. Quantummode.com
- 13)** J. Fernández (2014), Bases de datos NoSQL II, El modelo de datos. Quantummode.com
- 14)** J. Fernández (2014), Bases de datos NoSQL III, NoSQL en la práctica. Quantummode.com
- 15)** J. Fernández (2014), Bases de datos NoSQL IV, Las debilidades de NoSQL. Quantummode.com
- 16)** Cristian Requena (2010), Que es NoSQL, NoSQL.es. España
- 17)** Af83 (2011). The Quest for Extreme Scalability.
- 18)** Enrique López Albores (2013), fichas bibliográficas APA, UPAEP. Puebla, México.
- 19)** Miguel Orquera (2012), Almacenamiento en base de datos, Universidad Técnica del Norte. Imbabura – Ibarra – Ecuador
- 20)** Leandro Alegsa (2011), Definición de Almacenamiento de bases de datos
- 21)** Enrique Dans (2011), Big Data: una pequeña introducción. Universidad Santiago de Compostela. España. (Dans, 2011).
- 22)** Enrique Dans (2011), Entender el futuro: la evolución de las bases de datos. Universidad Santiago de Compostela. España.
- 23)** Carlos Andrés López Peña (2012), Análisis de las bases de datos NoSQL como alternativa a las bases de datos SQL. Escuela de Ingeniería de Antioquia. Colombia.
- 24)** Diana Marisela Brito Zhunio (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.  
Universidad Tecnológica Israel, Ecuador.

- 25)** Rosa Fernanda Córdova Espinoza, Bernardo Esteban Cuzco Sarango (2013), Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales. Universidad Politécnica Salesiana. Ecuador.
- 26)** Sergio Eduardo Mancilla Escobar (2013), uso de bases de datos NoSQL documentales para crear sitios web de alto rendimiento. Universidad de San Carlos de Guatemala, Guatemala.
- 27)** Aurelio Morales (2014), Sobre MongoDB, bases de datos NoSQL y GIS. Formación GIS y difusión tecnológica, España.
- 28)** Juan Carlos García Candela (2010) Bases de datos NoSQL y escalabilidad horizontal. Universidad de Alicante.
- 29)** Victoria Malaya (2013) SQL vs NoSQL
- 30)** Nick Salloum (2015) CRUD (Create, Read, Update, Delete)
- 31)** Meera Prince (2014) MongoDB NoSQL Database / Document Database
- 32)** José Manuel Alarcón (2014) Fundamentos de bases de datos SQL, inserción de datos, España.
- 33)** José Manuel Alarcón (2014) Fundamentos de bases de datos SQL, operaciones con conjuntos, España.
- 34)** José Manuel Alarcón (2014) Fundamentos de bases de datos NoSQL, MongoDB, España.
- 35)** Genbeta (2013) Bases de datos NoSQL. Elige la opción que mejor se adapte a tus necesidades, Mexico.
- 36)** Leonardo D (2013) ACID en las bases de datos, Argentina
- 37)** Francisco Xavier Ruano V (2014), Análisis y Desarrollo de MongoDB y Redis en Java, España.

## **CAPÍTULO VII**

### **ANEXOS**

## 7.1. Proceso de instalación en SQL.

MySQL es un motor de base de datos Open Source. Es usado en una gran cantidad de aplicaciones, cuenta con excelente documentación oficial y no oficial y es una de las grandes opciones a elegir como base de datos personal.

- Inicie Sesión en su computadora con una cuenta de administrador.
- Verifique que su pc, cumpla con los requisitos previos a la instalación. Adjunto enlace <https://msdn.microsoft.com/es-es/library/ms143506.aspx>
- Adicionalmente necesitamos disponer del programa de instalación. Se puede descargar gratuitamente de <http://dev.mysql.com/downloads> o desde la página web de la asignatura.

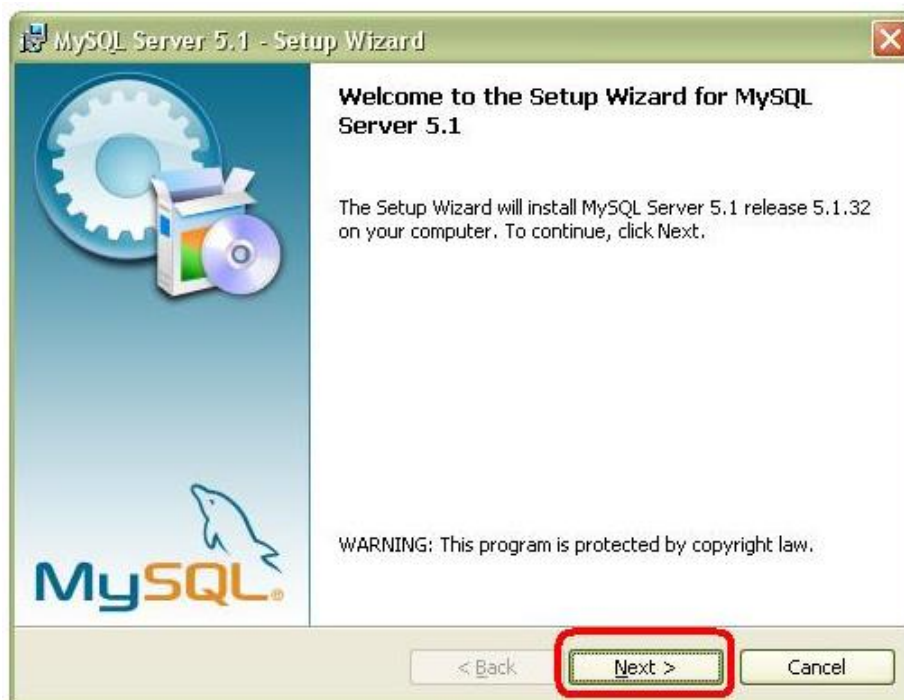
Una vez descargado el programa de instalación de MySQL lo ejecutaremos y seguiremos las instrucciones que nos muestra el asistente de instalación:

**Gráfico 25. Preparación proceso de instalación de MySQL**



- Aparece la pantalla de inicio del instalador. Hacemos clic en siguiente para continuar con el proceso de instalación.

**Gráfico 26. Pantalla de Instalación de MySQL**



- Seguidamente seleccionamos el tipo de instalación en nuestro caso seleccionamos instalación completa y damos clic en “siguiente”.

**Gráfico 27. Ventana de selección dl tipo de instalación en el equipo**



- Y damos clic en instalar, para proceder a realizar la copia de los archivos

**Gráfico 28. Ventana para proceder a realizar la copia de archivos comunes**



- Esperamos mientras se realiza la instalación de los archivos. A continuación hacemos clic en “Siguiente” en las pantallas de MySQL Enterprise.
- En la siguiente pantalla seleccionamos la opción “Configuración de nuevo servidor MySQL”

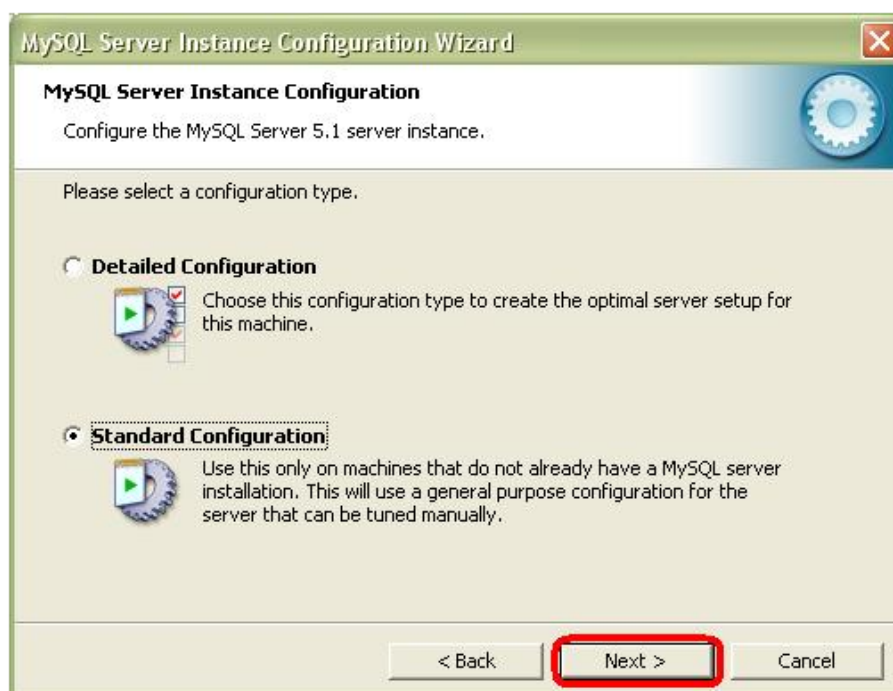


**Gráfico 29. Ventana de selección para proceder a configurar MySQL**



- A continuación procedemos a realizar la instalación de la instancia del MySQL, y seleccione la opción “Configuración Estándar” para su instalación y presione “Siguiente”.

**Gráfico 30. Ventana para seleccionar el tipo de configuración**



- Seleccione las opciones “Instalación de Nuevo servicio de Windows”, también “Incluir directorio Bin en Windows Path”. No siempre se desea que

el motor de base de datos arranque automáticamente cuando inicia el sistema operativo, para eso deseleccione la opción “Ejecutar MySQL Server automáticamente”, si lo hace, siempre deberá iniciar el servidor manualmente. No hay problema si se deja seleccionada la opción. Presione “Siguiente” para continuar.

**Gráfico 31. Instalación de MySQL como servicio de Windows**



- Introduciremos la contraseña para el usuario administrador (root) y marcaremos la opción "Habilitar acceso root desde Escritorio Remoto" si queremos que se pueda acceder como administrador desde otros equipos:

**Gráfico 32. Creación de nuevas credenciales de autenticación**

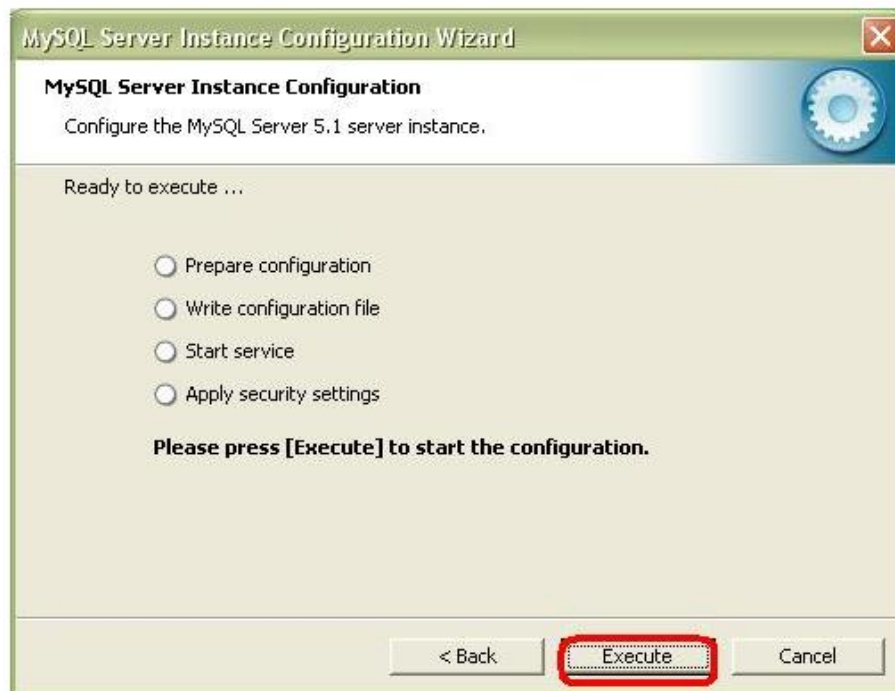


- Y para finalizar hacemos clic en “Ejecutar” para realizar la configuración. Si previamente tuvo instalado MySQL es probable que tenga problemas al iniciar el servicio, la mejor solución es usar la instalación previa para eliminar la instancia y desinstalar el producto completamente.

Sin embargo esto no siempre se puede hacer ya sea porque se desinstaló sin eliminar la instancia o porque el producto falló. En estos casos, se puede solucionar de la siguiente forma:

- Borrar la carpeta C:\Documents and Settings\<usuario\_actual>\Datos de programa\MySQL. Borrar la carpeta C:\Archivos de programa\MySQL.

**Gráfico 33. Ejecución del proceso de instalación de MySQL como servicio de Windows**

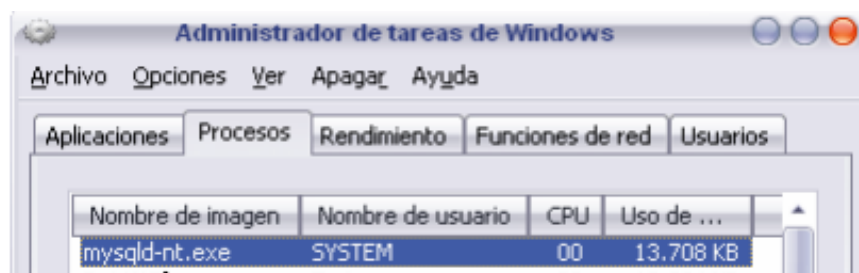


**Gráfico 34. Instalación Completa de MySQL como servicio Windows**



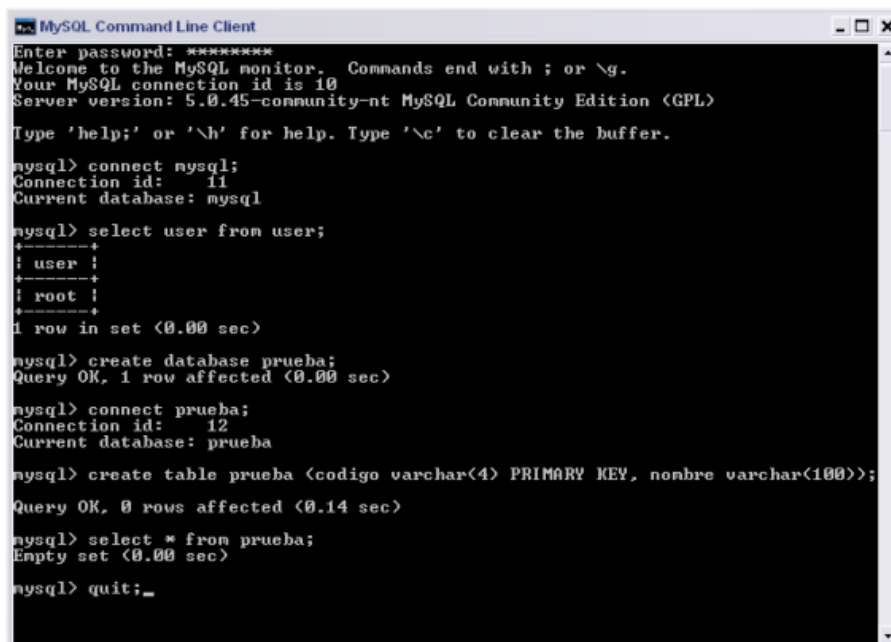
- Tras realizar la instalación podremos comprobar (si hemos seleccionado la opción de iniciar MySQL como servicio) que el servicio se está ejecutando. Esto se puede ver en el administrador de tareas:

**Gráfico 35. Visualización de MySQL como servicio de Windows**



- Nos aparecerá un servicio con el nombre "mysqld-nt.exe" que, como se puede observar, usa unas 12 MB de memoria RAM (sin conexiones de clientes). Se sugiere instalar las herramientas gráficas (GUI Tools) disponibles en la página web de la asignatura.
- Si lo deseamos podemos volver a configurar la instancia de MySQL desde "Inicio" - "Programas" - "MySQL" - "MySQL Server 5.0" - "MySQL Server Instance Config Wizard". El asistente que aparecerá será similar al explicado en el programa de instalación.

**Gráfico 36. En la siguiente imagen se aprecia la consola MySQL**



- También podremos configurar mediante la línea de comandos MySQL, para ello iremos a "Inicio" - "Programas" - "MySQL" - "MySQL Server 5.0" - "MySQL Command Line Client". Nos pedirá una contraseña (la que hayamos introducido en la instalación).

- Como ejemplo para comprobar que la instalación ha sido correcta nos hemos conectado a la base de datos que MySQL crea automáticamente llamada "mysql", la cual contiene los usuarios y configuración de MySQL.
- Para comenzar a utilizar MySQL seleccione la opción del menú inicio MySQL → MySQL Server 5.1 → MySQL Command Client. Ingrese la contraseña que utilizó en el paso 15 de la instalación.

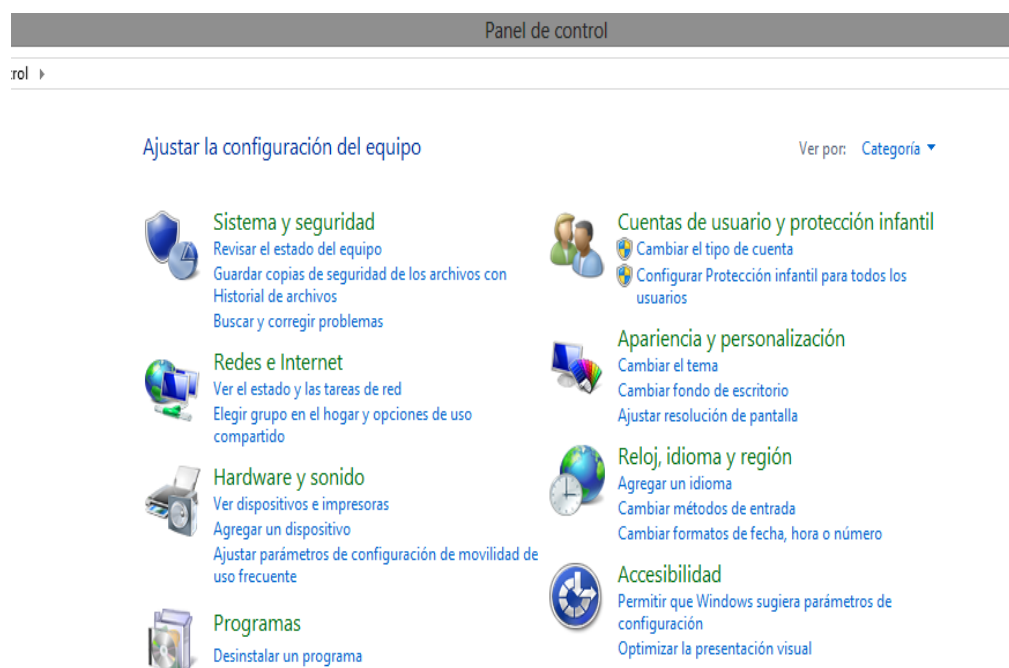
**Gráfico 37. Pantalla de ingreso a la consola MySQL**



Para permitir que el tráfico SQL fluya a través del firewall, los pasos a seguir son los siguientes:

- Ir al panel de control y arrancar la gestión del firewall.

**Gráfico 38. Pantalla del Panel de Control.**



**Gráfico 39. Selección de Sistemas y Seguridad para configuración del Firewall**

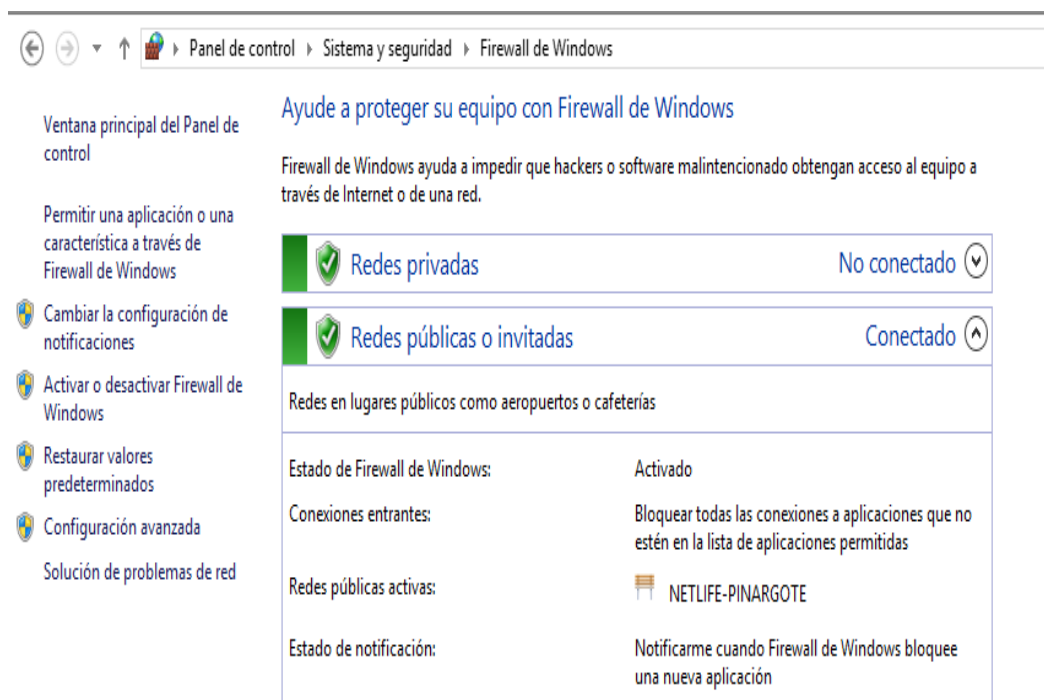


**Ilustración 41.**

- Acceder al apartado “Permitir un programa o una característica a través del firewall de Windows”.

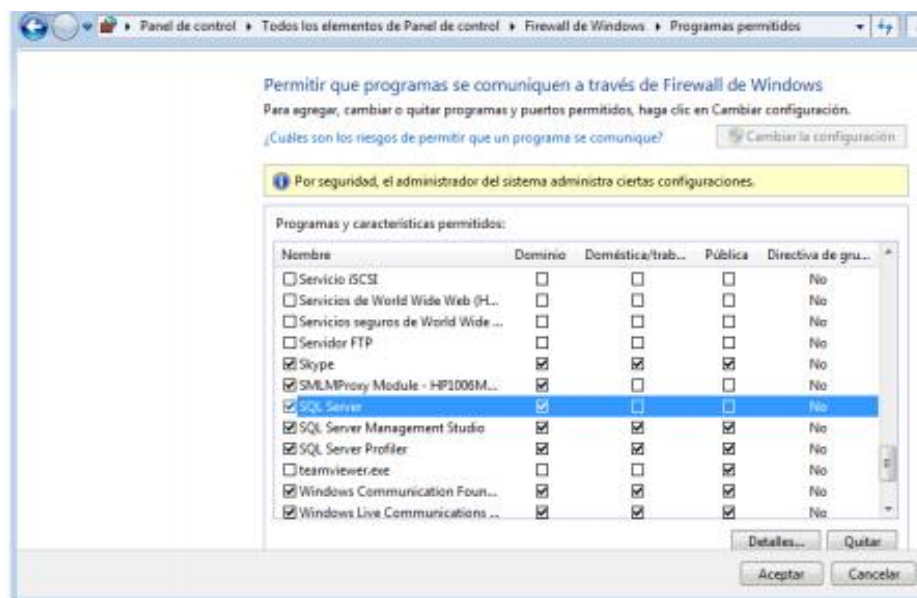


## Gráfico 40. Selección de Características de Windows



- En la siguiente pantalla, buscar la aplicación “SQL Server”. Activar el check “Dominio” o “Domestica/Trabajo (privada)”, según el tipo de red establecida.

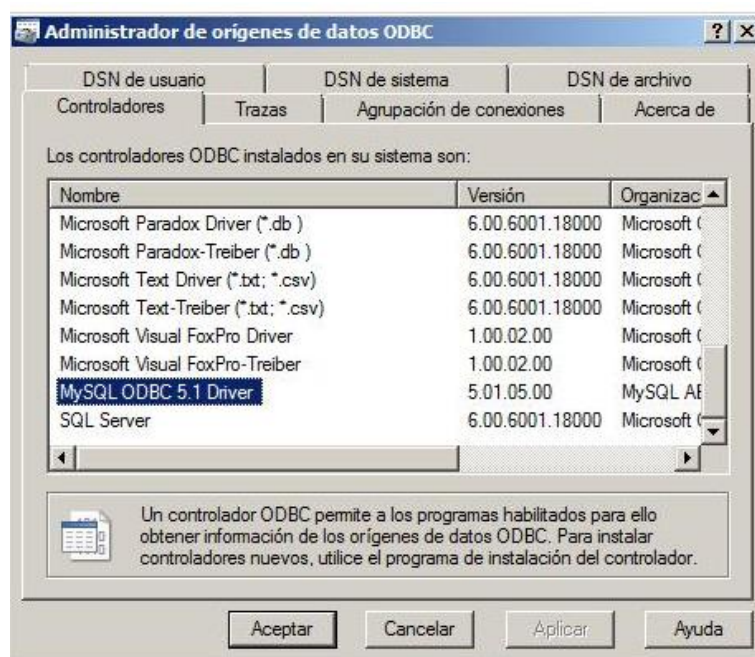
## Gráfico 41. Pantalla de activación de características de Windows



- Continuamos con el proceso y se recomienda, para que se efectúen los cambios en el sistema se reinicie el servicio desde el SQL Server Configuration Manager



**Gráfico 42. Visualización de MySQL como servicio de Windows**



Para mayor información sobre SQL Server 2012 puede redirigirse a ([www.microsoft.com](http://www.microsoft.com)), ya que es el ente encargado de desarrollar y distribuir la aplicación.

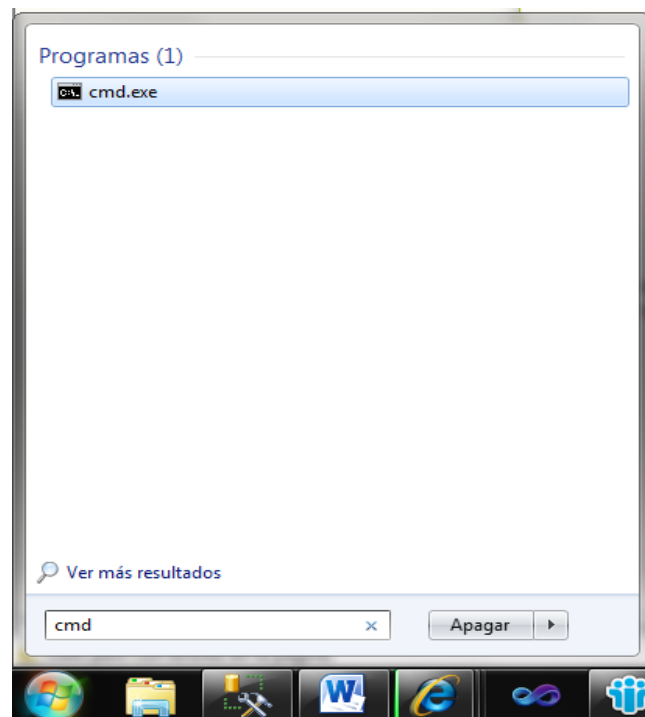
## **7.2. Proceso de instalación en NoSQL.**

### **Instalación MongoDB**

A continuación se describe el proceso de Instalación del gestor de almacenamiento de bases de datos MongoDB, probablemente una de las más famosas dentro de la familia NoSQL (Bases de datos No Relacionales).

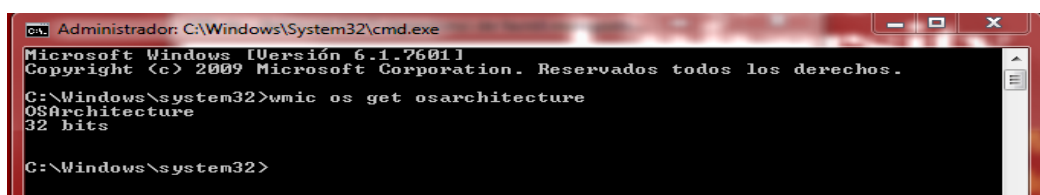
- Ejecutamos como administrador el comando cmd del sistema.

**Gráfico 43. Menú Inicio comando cmd.exe**



- Antes de iniciar la instalación necesitamos verificar que versión de MongoDB usaremos (x32 o x64 bits) para ello ingresamos una línea de comandos en nuestro Windows y usaremos el comando “**wmic os get osarchitecture**” para verificar con que versión se encuentra configurado en nuestro equipo.

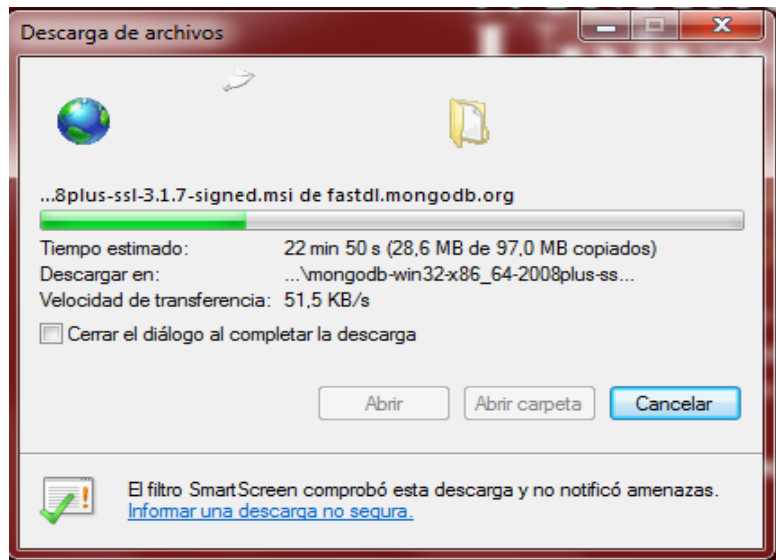
**Gráfico 44. Consola en modo Administrador**



- Si no cuenta con la versión que requiere el sistema puede descargarla desde el siguiente enlace, (<http://www.mongodb.org/downloads>) totalmente gratuito.
- Para todas las plataformas se encuentra tanto la versión de **32-bits como la de 64-bits**. Se debe escoger una u otra según la arquitectura de nuestro servidor u ordenador. En este punto es recomendable mencionar que **la versión de 32-bits tiene algunas limitaciones**, como que el tamaño de la base de datos no puede exceder de **2GB**, por lo que se recomienda su uso

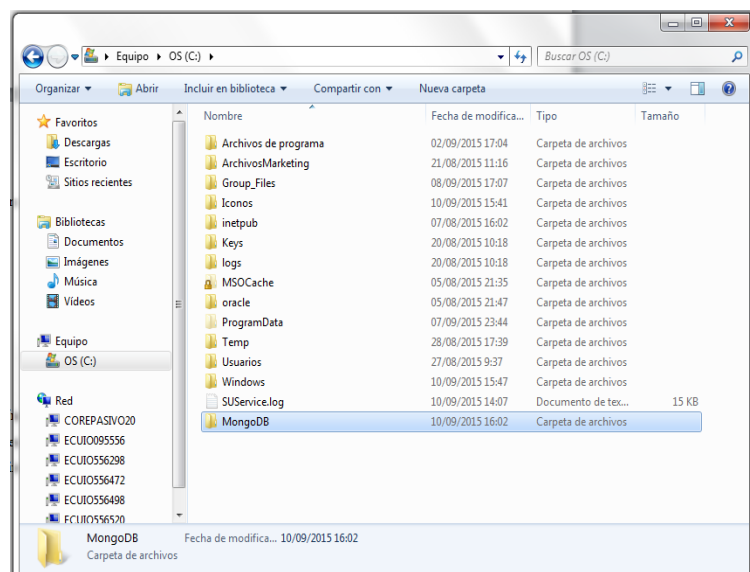
sólo para entornos de desarrollo y pruebas pero no para entornos reales o de producción.

**Gráfico 45. Proceso de descarga del motor de base de datos MongoDB versión 3.0.7**



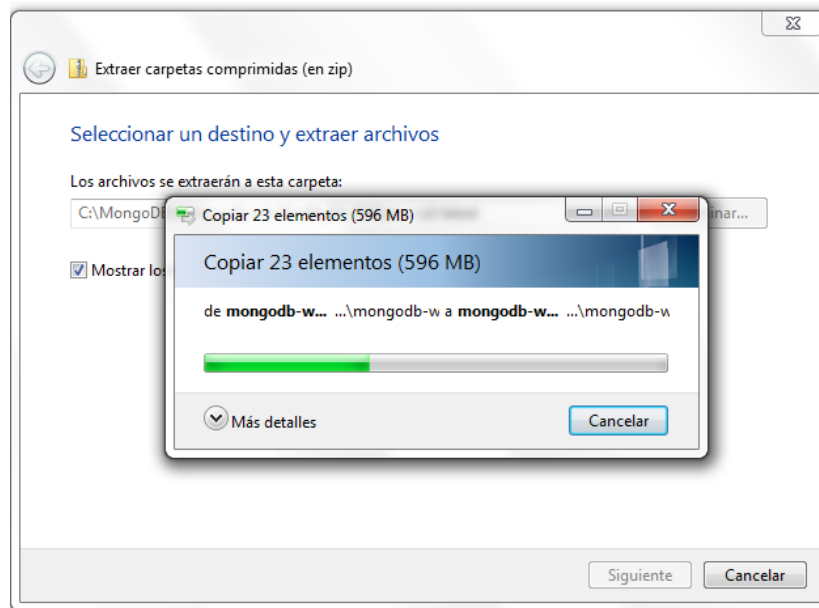
- Una vez descargado el archivo de instalación “**mongodb-win32-i386-3.0.7.zip**” lo descomprimiremos en un directorio en mi caso llamado “MongoDB” creado en el directorio raíz de la unidad C:\ para posteriormente realizar la instalación.

**Gráfico 46. Carpeta destino de la instalación de MongoDB**



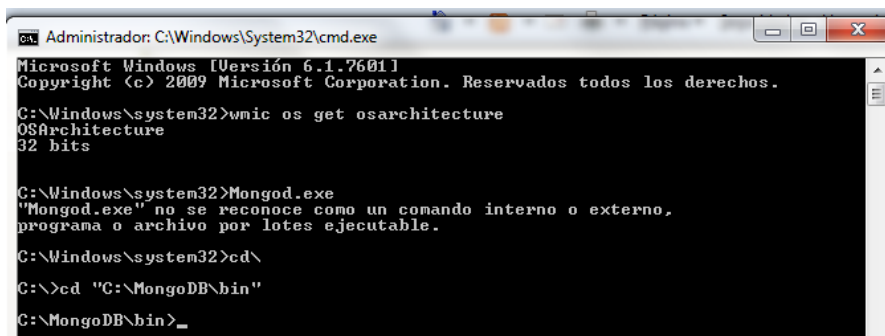
- Una vez descargado el archivo de instalación comprimido en Zip, procedemos a descomprimirlo en el directorio “MongoDB”

**Gráfico 47. Proceso de descarga del motor de base de datos MongoDB versión 3.0.7**



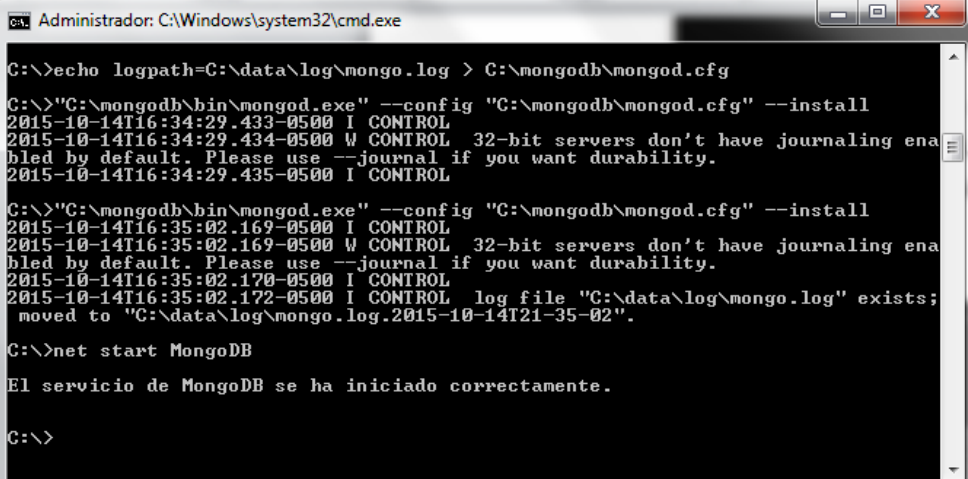
- Para almacenar los flujos de datos, creamos una nueva ruta "c:\data\db" que es el directorio por defecto que usa Mongo DB para trabajar.
- La creación del árbol de directorios se puede hacer como hemos explicado anteriormente para su posterior uso en la creación del directorio "MongoDB" o lo podemos crear mediante el uso de comandos.

**Gráfico 48. Instalación y configuración de MongoDB**



- A continuación ejecutamos el fichero de instalación de MongoDB, de la siguiente manera, accedemos mediante comandos al directorio de instalación y tecleamos "**Mongod.exe**" dentro del **cmd** en calidad de administrador y presionamos "**Enter**", nos mostrará el siguiente cuadro.

**Gráfico 49. Proceso para iniciar el servicio MongoDB**



```
Administrador: C:\Windows\system32\cmd.exe

C:\>echo logpath=C:\data\log\mongo.log > C:\mongodb\mongod.cfg

C:\>"C:\mongodb\bin\mongod.exe" --config "C:\mongodb\mongod.cfg" --install
2015-10-14T16:34:29.433-0500 I CONTROL
2015-10-14T16:34:29.434-0500 W CONTROL 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2015-10-14T16:34:29.435-0500 I CONTROL

C:\>"C:\mongodb\bin\mongod.exe" --config "C:\mongodb\mongod.cfg" --install
2015-10-14T16:35:02.169-0500 I CONTROL
2015-10-14T16:35:02.169-0500 W CONTROL 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2015-10-14T16:35:02.170-0500 I CONTROL
2015-10-14T16:35:02.172-0500 I CONTROL log file "C:\data\log\mongo.log" exists; moved to "C:\data\log\mongo.log.2015-10-14T21-35-02".

C:\>net start MongoDB

El servicio de MongoDB se ha iniciado correctamente.

C:\>
```

- Este procedimiento instala MongoDB con los parámetros por defecto, al iniciar la instalación es posible recibir un aviso del firewall de Windows diciéndonos que se han bloqueado algunas características de Mongo, mostrándonos un mensaje.

En el ese caso de que esto ocurriese daremos clic en el botón de “Permitir Acceso” para continuar. En ese momento terminaremos nuestra instalación y tendremos a MongoDB funcionando como servicio de Windows, para iniciar el servicio usaremos el comando "Net start MongoDB", y para detener el servicio usaremos el comando “Net stop MongoDB”