

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA

CARRERA DE INGENIERÍA DE SISTEMAS

**“ANÁLISIS COMPARATIVO ENTRE BASES DE DATOS
RELACIONALES CON BASES DE DATOS NO RELACIONALES”**

**Tesis previa a la obtención del Título de:
Ingeniero en Sistemas**

AUTORES

Rosa Fernanda Córdova Espinoza

Bernardo Esteban Cuzco Sarango

DIRECTOR

Ing. Germán Parra González

CUENCA – ECUADOR

2013

CERTIFICACIÓN

Certifico que el presente trabajo de tesis previo a la obtención del título de Ingeniero de Sistemas fue desarrollado por Rosa Fernanda Córdova Espinoza y Bernardo Esteban Cuzco Sarango bajo mi supervisión.

A handwritten signature in blue ink, appearing to read 'G. Parra', is written over a horizontal dotted line.

Ing. Germán Parra González

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Rosa Fernanda Córdova Espinoza y Bernardo Esteban Cuzco Sarango, estudiantes de la carrera de Ingeniería de Sistemas declaramos que los conceptos, análisis desarrollados y conclusiones realizadas, son de nuestra responsabilidad y autoría en la Tesis: “ANÁLISIS COMPARATIVO ENTRE BASES DE DATOS RELACIONALES CON BASES DE DATOS NO RELACIONALES”, y autorizamos a la Universidad Politécnica Salesiana el uso de la misma con fines académicos.

A través de la presente declaración cedemos los derechos de Propiedad Intelectual correspondiente a este trabajo a la Universidad Politécnica Salesiana, según lo establecido por la ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.



.....
Rosa Fernanda Córdova Espinoza



.....
Bernardo Esteban Cuzco Sarango

Dedicatoria

Dedico esta tesis a DIOS, por darme la vida, salud y las fuerzas necesarias para no desmayar y seguir adelante frente a los problemas que se me presentaron, permitiéndome llegar a este momento tan importante de mi formación académica.

A mis padres que gracias a su apoyo, aliento, amor, ayuda en los momentos difíciles supieron darme todo, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis propósitos sin lastimar a nadie, a pesar de las diferencias que teníamos, depositaron su entera confianza en cada reto sin dudar ni un solo momento en mi inteligencia y capacidad.

A mis hermanos y hermanas a pesar de no tenerles a todos a mi lado, gracias por apoyarme, motivarme y aportar en mí la lucha necesaria para poder alcanzar este sueño.

A mi hijo Mateo quien ha sido y será el motor más importante en mi vida, por demostrarme cada día su amor sin condiciones y quien estuvo todo este tiempo a mi lado de quien me he robado sus sueños, su tiempo, gracias mi amor chiquito.

A mis compañeros con quienes he realizado mis trabajos y tareas.

A mis amigos con los que he compartido y he pasado momentos inolvidables, con quienes he llorado, reído, sufrido gracias.

Fernanda

Agradecimiento

En primer lugar agradezco a DIOS por darme la vida, por guiarme por el camino de la felicidad y por haberme permitido llegar al final de mi carrera.

Agradezco a mi familia, mis padres, hermanos, sobrinos, quienes supieron darme el ejemplo de perseverancia en mis cosas en luchar por mis sueños en hacerlo de corazón entregando todo de uno y con una sonrisa en los momentos difíciles.

A mi hijo Mateo te agradezco por entenderme, por comprender que había momentos que no podía pasar a tu lado, TE AMO.

Agradezco a todos mis profesores quienes me han acompañado durante todo este tiempo, a quienes les debo gran parte de mis conocimientos, quienes me brindaron sus consejos y me orientaron con profesionalismo para mi formación correcta.

Agradezco de manera especial y sincera a mi director de tesis Ing. Germán Parra por aceptar la dirección de la misma, quien con su apoyo, confianza, paciencia me ha orientado en todo momento y ha sido la clave del buen trabajo que hemos realizado juntos. Destacando por encima de todo su disponibilidad en todo momento y que han significado para una solida amistad, muchas gracias Ingeniero.

Agradezco a Luis por estar conmigo en los momentos difíciles y quien siempre me apoyo sin esperar nada a cambio.

A mi compañero de tesis Bernardo con quien he pasado momentos buenos y malos, pero siempre hemos salido adelante y hemos logrado sacar adelante este proyecto.

Fernanda

Dedicatoria

A PAPA DIOS

Por brindarme grandes pasos de esfuerzo y perseverancia en los momentos donde necesitaba de su fortaleza, sabiduría y entendimiento para lograr el fin alcanzado.

A MI MAMA LUCIA.

Por tenerme paciencia, por sus buenos consejos de superioridad, su ayuda idónea en todo momento a nivel personal, a nivel académico, y siendo más que una madre una amiga sin igual, en donde puedo depositar toda mi confianza.

A MI PAPA ALFONSO

Como ejemplo de un buen hombre, que a pesar de las circunstancias, no ha retrocedido en su forma de ser.

A MIS FAMILIARES

A mi hermana Fabiola y su familia, toda mi admiración con el ejemplo de vida de una profesional, ahora madre, una mujer luchadora que se supera constantemente

A mi hermano Alfonso, un profesor de vida, que comparte sus conocimientos sin esperar nada a cambio.

A mi tía Ángela y mi primo Andrés, formando parte de una familia de un constante apoyo.

A mi abuelita Celestina, por su carisma y alegría que me imparte cada día, al comienzo o término de una jornada de labores.

A MIS AMIGOS

De una u otra forma, compartieron su tiempo para la realización de esta tesis, amigos/as donde se vivió momentos agradables, en reuniones, bailes. A todos ellos que nos supieron apoyar sin interés alguno.

Bernardo

Agradecimiento

A Papa Dios, gracias por hacerme tu hijo, donde tú me creaste y formaste a tu imagen desde antes de nacer, a mis líderes tanto Héctor, Isabel, Carolina y Lorena que han dispuesto de su tiempo para enseñarme principios de vida.

A mis padres y hermanos que me dieron su apoyo incondicional en los momentos que más los necesite, y a toda mi familia.

Al Ingeniero Germán Parra, donde con su apoyo, comprensión, paciencia; nos guió, apoyo en las dificultades para la terminación de esta etapa académica.

Bernardo

ÍNDICE DE CONTENIDO

CAPÍTULO 1	5
USO DE LAS BASES DE DATOS RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES	5
1.1 INTRODUCCIÓN	5
1.2 CONCEPTOS GENERALES	6
1.2.1 DEFINICIÓN DE BASE DE DATOS RELACIONALES	6
1.2.2 OTROS TÉRMINOS	6
1.3 TIPOS DE SISTEMAS DE GESTIÓN DE BASE DE DATOS	7
1.3.1 SISTEMAS PROPIETARIOS	10
1.3.2 SISTEMAS DE CÓDIGO ABIERTO	18
1.4 GENERALIDADES DE LAS BASES DE DATOS RELACIONALES	25
1.4.1 FUNCIONES DE UTILIZAR UNA BASE DE DATOS RELACIONAL 25	
1.4.2 BENEFICIOS DE UTILIZAR UNA BASE DE DATOS RELACIONALES	25
1.4.3 VENTAJAS DE UNA BASE DE DATOS RELACIONAL	26
1.4.4 DESVENTAJAS DE UNA BASE DE DATOS RELACIONAL	26
CAPÍTULO 2	27
DESVENTAJAS DEL USO DE BASE DE DATOS RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES	27
2.1 ANÁLISIS DE LAS BASES DE DATOS GNU	27
2.2 ANÁLISIS DE BASE DE DATOS COMERCIALES	28
2.3 DESVENTAJAS	30
CAPÍTULO 3	32
ESTUDIO DE BASE DE DATOS NO RELACIONALES	32
3.1 INTRODUCCIÓN	32
3.2 CONCEPTOS GENERALES	34
3.2.1 DEFINICIÓN DE NoSQL	34

3.3 TIPOS DE BASES DE DATOS NO RELACIONALES	35
3.3.1 ALMACENES KEY - VALUE (CLAVE - VALOR)	36
3.3.2 ORIENTADAS A COLUMNAS	41
3.3.3 BASADA EN DOCUMENTOS.....	42
3.3.4 ORIENTADAS A GRAFOS	44
3.4 GENERALIDADES DE LAS BASES DE DATOS NO RELACIONALES.	45
3.4.1 FUNCIONES	45
3.4.2 BENEFICIOS.....	46
3.4.3 VENTAJAS DE LAS BASES DE DATOS NO RELACIONALES	46
3.4.4 DESVENTAJAS DE LAS BASES DE DATOS NO RELACIONALES ..	47
ANÁLISIS DE BASES DE DATOS NO RELACIONALES Y RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES.	55
4.1 IMPLEMENTACIÓN DE UNA APLICACIÓN WEB	55
4.2 NORMA ISO/IEC 9126	60
4.3 PRUEBAS DE RENDIMIENTO	67
4.4 COMPARACIÓN DE BASE DE DATOS NO RELACIONALES VS BASES DE DATOS RELACIONALES	69
4.5 RESULTADOS.....	69
4.5.1 MATERIALES Y CONDICIONES CON QUE SE REALIZARON LAS PRUEBAS DE RENDIMIENTO	70
CAPÍTULO 5	74
CONDICIONES PARA EL USO DE LAS BASES DE DATOS NO RELACIONALES.	74
5.1 NORMATIVAS	74
CONCLUSIONES	77
RECOMENDACIONES.....	78
BIBLIOGRAFÍA	79
ANEXOS	80

ÍNDICE DE FIGURAS

Figura 1: Partes de la Arquitectura de Oracle	10
Figura 2: Estructuras de un Servidor Oracle	13
Figura 3: Capa Física	15
Figura 4: Capa Lógica.....	16
Figura 5: Arquitectura de MySQL.....	20
Figura 6: Almacenes Key – Value	36
Figura 7: Esquema de una Base de Datos Relacional.....	51
Figura 8: Esquema de MongoDB.....	51
Figura 9: Teorema de CAP	75

ÍNDICE DE TABLAS

Tabla 1: Top Ten de renovación de Licencias de la IDC y db-engines.com	9
Tabla 2: Almacenamiento de las Tablas	24
Tabla 3: Base de Datos Orientada a Columnas	41
Tabla 4: Base de Datos Orientada a Objetos.....	42
Tabla 5: Base de Datos Relacional (estructura de los datos)	43
Tabla 6: Características de las Bases de Datos	45
Tabla 7: MongoDB vs Base de Datos Relacional.....	50
Tabla 8: Versiones de Software	56
Tabla 9: Versiones de Software	59
Tabla 10: Comparación Base de Datos Relacional vs No Relacionales	69
Tabla 11: Características del Servidor	70
Tabla 12: Características Cliente 1	70
Tabla 13: Características Cliente 2	70
Tabla 14: Tamaño de las Bases de Datos.....	71
Tabla 15: Resultados con 49.545 registros	71
Tabla 16: Resultados con 125.254 registros	72

CAPÍTULO 1

USO DE LAS BASES DE DATOS RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES

1.1 INTRODUCCIÓN



1

Ilustración 1: Introducción Base de Datos

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 permiten 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 las conocemos, comenzando con el almacenamiento de información con tarjetas perforadas y cintas magnéticas, uso de discos que dieron inicio a las bases de datos, bases de datos de red y bases de datos jerárquicas. Surge un modelo relacional, que un principio consistió en una serie de reglas para la evaluación de administradores de sistemas de datos relacionales.

¹Imagen tomada de: <http://www.niux.com.ar/bases-de-datos/>

Este modelo tuvo poca acogida sobre todo por tener un menor rendimiento que otros sistemas de almacenamiento. Pues estos sistemas utilizaban tablas para almacenar información, excluyendo de elementos importantísimos como son de claves primarias, etc.

1.2 CONCEPTOS GENERALES

1.2.1 DEFINICIÓN DE BASE DE DATOS RELACIONALES

Es un modelo de datos utilizado para modelar problemas reales y administrar datos dinámicamente, su idea fundamental son las relaciones que se dan entre las entidades del diagrama como conjuntos de datos llamados tuplas, la mayoría de veces se conceptualiza de una manera fácil de imaginar, dándose cuenta que cada relación fuese una tabla compuesta por registros representando las tuplas y los campos.²

1.2.2 OTROS TÉRMINOS

Sistema Gestor de Base de Datos (SGBD)

Es un conjunto de datos interrelacionados y con herramientas computacionales específicas para acceder a dichos datos. Su lugar de almacenamiento se denomina base de datos, pues es aquella que contiene información relevante de una empresa u organización. El objetivo principal de un SGBD es de almacenar y recuperar la información de la organización o empresa de una forma práctica y eficiente.³

² Wikipedia – 27 de febrero del 2012.- http://es.wikipedia.org/wiki/Modelo_relacional

³SILBERSCHATZ, Abraham, Fundamentos de Bases de Datos, Cuarta Edición, 2002, p, 1

Modelo de Datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

Base de Datos Relacional

Es un repositorio compartido de datos. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre **Modelo Relacional**, este tipo de base de datos es el más utilizada en la actualidad para implementar aplicaciones.⁴

1.3 TIPOS DE SISTEMAS DE GESTIÓN DE BASE DE DATOS

Existen sistemas de gestión propietarios y sistemas de código abierto, entre los sistemas de gestión propietarios tenemos:

- Oracle
- Microsoft SQL Server
- DB2, etc.

Sistemas de gestión de código abierto como:

- MySQL
- Postgres
- Firebird
- SQLite, etc.

⁴ SILBERSCHATZ, Abraham, Fundamentos de Bases de Datos, Cuarta Edición, 2002, p, 5

Razones porque se eligieron Oracle y MySQL como base de datos de estudio

Para determinar los sistemas de bases de datos a analizar hemos considerado los siguientes elementos: precio de renovación de licencias, la simplicidad de estos sistemas tanto en hardware como software y su eficiencia. Cabe mencionar que no existe un sistema de base de datos perfecta, cada una de estas tiene sus características y deficiencias.

Según un estudio realizado por IDC⁵ análisis en el año del 2009, los 10 sistemas de bases de datos a considerar tenemos, a Oracle como primer puesto, seguido de SQL Server de Microsoft y posteriormente de Db2 de IBM, con respecto a sistemas de código abierto tenemos a MySQL como primera opción seguido de Postgres entre otras, concluye que Oracle sigue siendo el RDBMS más utilizado y adquirido para los diferentes sistemas operativos.

Mientras que la db-engines.com⁶ vemos que siguen liderando las bases de datos relacionales SQL, en el primer puesto está Oracle, SQL Server y MySQL se van alternando el segundo puesto, algo lejos quedan las bases de datos como Access o la otra propuesta del software libre, postgresSQL, como se muestra en la tabla 1:

⁵ IDC (*International Data Corporation*): empresa líder en inteligencia de mercado, consultoría y eventos en las industrias de tecnología de la información, telecomunicaciones y mercados de consumo masivo de tecnología.

⁶ DB-Engines: es una iniciativa para recoger y presentar información sobre los sistemas de gestión de base de datos (DBMS).

	2013 (db-engines.com)	2008 – 2009 (IDC)
1	Oracle	Oracle Corp.
2	Microsoft SQL Server	Microsoft
3	MySQL	IBM
4	PostgreSQL	Teradata
5	DB2	Sybase
6	Microsoft Access	Fujitsu
7	MongoDB	Netezza
8	Sybase	Progress
9	SQLite	Software
10	Teradata	SAS Institute

Tabla 1: Top Ten de renovación de Licencias de la IDC y db-engines.com

Oracle es un sistema de gestión propietario utilizado en varias empresas, pues las mismas las usan como plataforma base ya que provee de un DBMS completo, su administración de la información es rápida y eficiente, esto es muy importante para las empresas porque necesitan que sus consultas respondan de una manera óptima.

Algunas bases están limitados en el volumen de sus datos, cuando el tiempo de respuesta sobrepasa un límite determinado suele entenderse que el sistema no soporta una carga de datos muy grande.

En cambio tenemos a MySQL como un sistema de gestión de código abierto, cuya característica principal es de su arquitectura, ya que está formado por dos capas, la primera una capa de SQL y la otra capa sus motores de almacenamiento; mientras que otros sistemas se encuentran unificadas, es decir con un solo motor de almacenamiento.⁷

⁷ <http://www.serverwatch.com/print.php/3883441>

1.3.1 SISTEMAS PROPIETARIOS

Oracle

Oracle es una empresa que se inicia en los años 70 por un estudio realizado por George Koch sobre los sistemas gestores de datos siendo una de las bases de datos más completas en soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma, basándose en una arquitectura cliente servidor para la gestión de bases de datos relacionales desarrollada por Oracle Corporación, con una interfaz comprensible, siendo capaz de administrar bases de datos, crear tablas, vistas y otros objetos de bases de datos, importar, exportar y visualizar datos de tablas, ejecutar scripts de SQL y generar informes.

Oracle tiene una arquitectura que se compone de dos partes, la **base de datos** que consta de las estructuras físicas y las estructuras lógicas y la otra parte de la **instancia de la base de datos** que se compone de estructuras de memoria y de procesos asociados con la instancia.

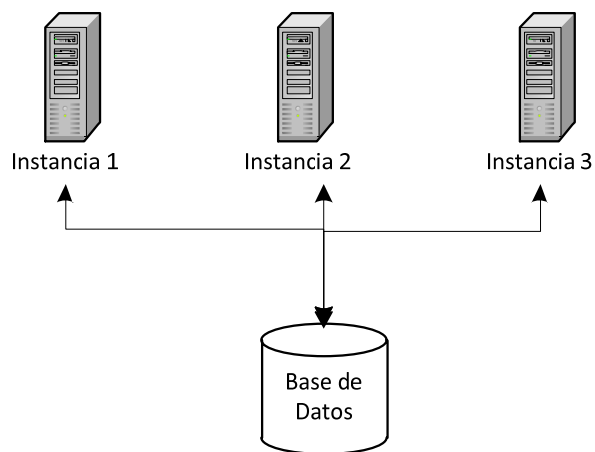


Figura 1: Partes de la Arquitectura de Oracle

Cuando se inicia una instancia se guarda un espacio de memoria y al mismo tiempo se arranca una serie de procesos, de esta manera el software de la base de datos es la que se encarga de asociarla con una base de datos específica la que le llamamos **montar la base de datos**. En ese momento dicha base está lista para manipularla y que los usuarios se conecten a ella, cada instancia está relacionado con una sola base de datos ya que esta no puede estar compartida por ninguna otra.

Instancia de la Base de Datos - Estructura de la Memoria

SGA (Área Global del Sistema)

Es la que se da al momento de iniciar una instancia siendo el componente fundamental de la misma, contiene:

- **Database Buffer Cache**

Es el área de trabajo de Oracle para la ejecución de los SQL, su tamaño se ajusta dinámicamente y su administración se da automáticamente.

- **Log Buffer**

Almacena los cambios realizados en la base de datos, la misma que se almacena en entradas de redo (estos contienen la información para reconstruir los cambios que se dieron en la base).

- **Shared Pool**

Es el cache de biblioteca (almacena el código ejecutado), cache del resultado de las consultas SQL y PLSQL, cache del diccionario de datos (almacena los objetos cuando las instrucciones han sido ejecutadas).

- **Large Pool**

Es opcional, se usa de forma automática por procesos, se usa normalmente para compartir los procesos del servidor.

- **Java Pool**

Es utilizado si se desea colocar código dentro del servidor.

- **Strem Pool**

Es un mecanismo utilizado para extraer vectores de cambio desde el online REDO LOG formando la reconstrucción de las consultas que se van a ejecutar.

PGA (Área Global de Programas)

Se da en el momento de iniciar el servidor, esta memoria es privada, siendo accesible a los procesos del servidor, por lo tanto no es compartida.

Procesos

- **De usuario**

Es el encargado de la interacción con el servidor realizando una conexión, para interactuar con el servidor se realiza a través de procesos.

- **De servidor**

Está encargado de satisfacer las llamadas generas devolviendo los resultados.

- **De fondo o segundo plano (background)**

Mantiene la relación de la base entre la memoria y el almacenamiento físico.

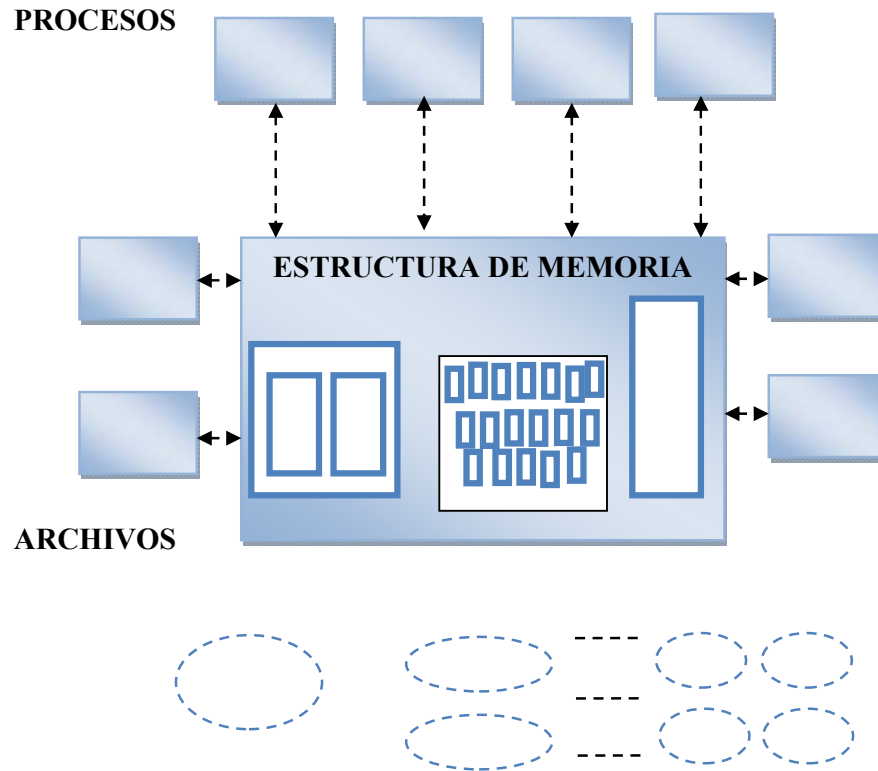


Figura 2: Estructuras de un Servidor Oracle

En el gráfico se muestra la relación de los procesos con las estructuras físicas y de la memoria de un Servidor Oracle

La Base de Datos de Oracle

La base de datos de Oracle consta de una capa física que consiste en un conjunto de archivos que se encuentran en el disco y una capa lógica son las estructuras que mapean los datos hacia los componentes físicos.

Capa Física

Archivos de Control

Contienen información que se utiliza al momento que se incrementa una instancia, la información de la ubicación de los datafiles y los archivos redo log, teniendo en cuenta que estos archivos deben estar protegidos.

Archivos de Datos

Los datafiles son los encargados de almacenar toda la información que es ingresada en la base de datos, pudiendo tener uno o cientos de ellos.

Muchas tablas e índices pueden compartir varios datafiles, los datafiles que son configurados están limitados por el parámetro de sistema MAXDATAFILES.

Archivos Redo Log

Estos archivos almacenan la información utilizada para la recuperación de una base de datos en caso de existir fallos, de igual manera almacenan todos los cambios realizados sobre la base de datos siendo útiles al momento de confirmar si los cambios que se han dado en la base de datos, confirmen que se han efectuado de igual manera en los datafiles.

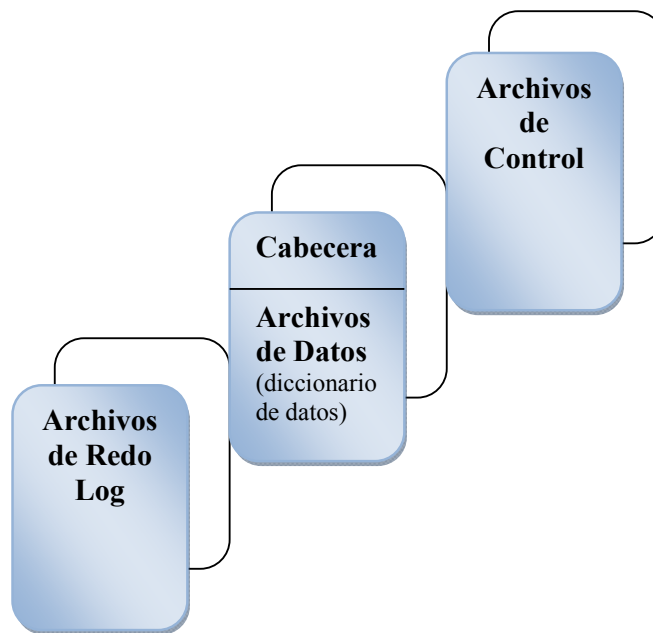


Figura 3: Capa Física

En la siguiente figura se muestran como los archivos de la capa física interactúan para el almacenamiento de la información de una manera correcta

Capa Lógica

Los tablespaces y los datafiles

La base de datos se encuentra dividida en una o varias piezas lógicas como los tablespaces que se encargan de separar la información en grupos para una correcta administración de los datos, pueden ocupar uno o más datafiles; al utilizar varios de estos el administrador del sistema puede gestionar la localización en los discos diferentes aumentando de esta manera el rendimiento para una mejor distribución de la carga de los datos.

El espacio que es utilizado para el almacenamiento de los datos es controlado por estructuras que se describen a continuación:

- **Bloques**

Es la unidad de almacenamiento más pequeña de una base de datos Oracle, contiene una porción de la información referente al bloque y a los datos que se guarda, ocupando aproximadamente 2 KB del espacio físico en el disco.

- **Extensiones**

Es un grupo de bloques de datos con un tamaño fijo creciendo a medida que se almacenan los datos, pudiendo redimensionar para aprovechar el espacio de almacenamiento.

- **Segmentos**

Es un grupo de extensiones que utiliza para almacenar datos particulares. Existen 4 tipos de segmentos: datos, índices, rollback y temporales.

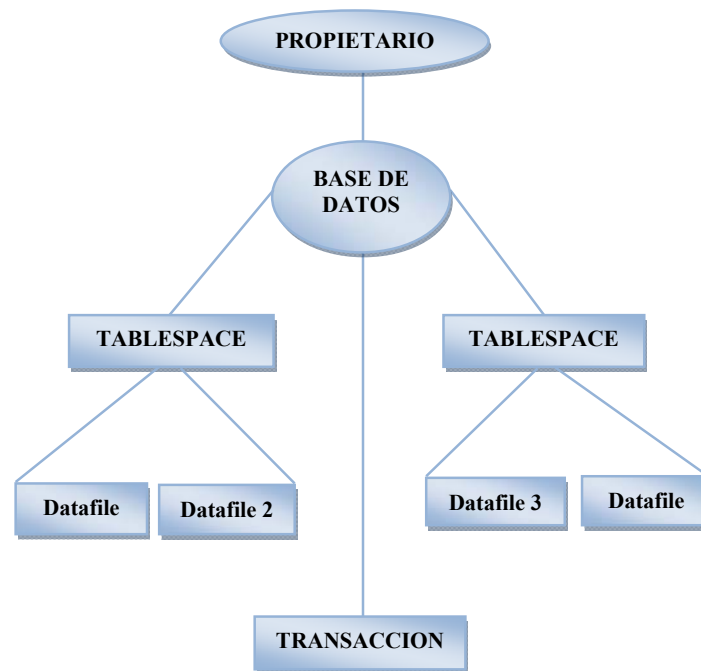


Figura 4: Capa Lógica

Como se muestra en la figura podemos ver como la capa lógica está dividida por varias piezas para que la información que se separa en diferentes grupos nos permita una correcta administración y una localización rápida en los discos

Estructura de la Base de Datos Oracle

Una base de datos de Oracle contiene varios tipos de archivos:

- **Archivos de Datos**

Este archivo contiene los datos actuales de la base de datos así como un diccionario de datos.

- **Archivos redo logs**

Almacena datos recuperables en caso de errores graves.

- **Archivos de Control**

Este archivo es necesario para mantener la integridad de la base de datos, también contiene la información del nombre de la base de datos, los archivos de datos, la información de la sincronización necesaria para una recuperación en un posible fallo. También es requerido para poder montar, abrir y acceder a la base de datos.

- **Archivos de parámetros**

Este archivo define algunas características de una instancia de Oracle.

- **Archivos de contraseñas**

Este archivo sirve para establecer las claves de usuarios administrativos y autenticar la entrada de los usuarios, también permite conectarse remotamente para realizar tareas administrativas.

- **Copias de archivos redo log**

Este archivo se utiliza para la recuperación de los datos.

Ventajas de Oracle

De acuerdo al criterio de Mayra Karina Cruz Jiménez podemos citar algunas ventajas y desventajas de Oracle:

- Oracle es el motor de base de datos relacional más usado a nivel mundial.
- Este se puede ejecutar en todas las plataformas.
- Oracle nos permite el uso de particiones para una mejor eficiencia en el momento de realizar una replicación de la base de datos este no pierda su integridad en sus datos.
- Es una base de datos con más orientación a Internet.

Desventajas de Oracle

- La mayor desventaja que tiene Oracle es en su precio, ya que algunas de sus licencias son demasíadamente caras.
- No tiene licencias libres.
- El costo en la información ya que su material, libros sobre los asuntos técnicos sobre su instalación y administración son difícil de encontrar libremente.⁸

1.3.2 SISTEMAS DE CÓDIGO ABIERTO

Dentro de los sistemas manejadores de bases de datos de código abierto, se presenta a continuación una descripción de MySQL:

⁸ <http://es.slideshare.net/Mayra00/taller-de-base-de-datos>

MySQL

Es una base de datos entre las más usadas en el mundo. Gracias a su arquitectura, es rápida, fácil de personalizar o configurar, su reutilización de código dentro del software ha logrado un sistema de administración con mayor velocidad, compactación, estabilidad y facilidad de despliegue, tiene una compañía comercial llamada MySQL AB, desarrollada por los fundadores de MySQL.

Es una base de datos relacional cuya función es de archivar los datos en tablas separadas en vez de colocar en solo archivo. Las mismas tablas están conectadas por relaciones que hacen posible combinar datos de diferentes tablas; con el fin de obtener mayor velocidad y flexibilidad.

Arquitectura de MySQL

La característica más notable que tiene MySQL es de separar el motor de almacenamiento (encargado de los detalles de entrada-salida, y de la representación de información en el disco duro) del resto de componentes de la arquitectura; permitiendo la posibilidad de crear nuevos motores para aplicaciones específicas.

Arquitectura Lógica de MySQL

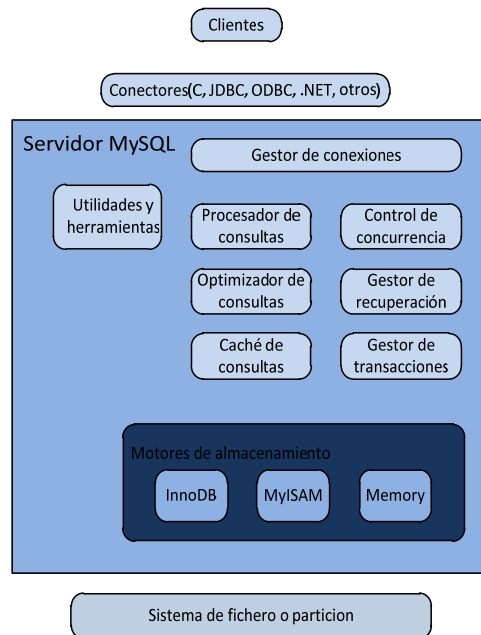


Figura 5: Arquitectura de MySQL

Como se muestra en el figura 6, los componentes del servidor se encuentran separadas, de igual manera las aplicaciones del cliente que lo utilizan y partes del sistema operativo

Entre las herramientas y utilidades tenemos las herramientas de backup, navegador de consultas, herramientas de diseño MySQL Workbench ⁹entre otras.

Motores de Almacenamiento de Datos

Lo más notable de la arquitectura de MySQL es que puede utilizar diferentes motores de almacenamiento para diferentes bases de datos o para tablas en la misma base de datos; permitiendo escoger el más adecuado para cada necesidad. Además permite que ciertos motores de almacenamiento implementen parte de los servicios

⁹ MySQL Workbench: herramienta visual de diseño de bases de datos, que integra el desarrollo de software, administración, diseño, creación y mantenimiento de bases de datos para el sistema de base de datos MySQL.

establecidos, obteniendo resultados óptimos en ciertas aplicaciones y deficientes en otras.

Entre los motores de almacenamiento de datos más usados destacan: MyISAM¹⁰ e InnoDB. Ninguno de estos motores se destaca una solución idónea para problemas específicos, ya que cada uno tiene sus ventajas y desventajas en campos específicos.

Con respecto a MyISAM, se encuentra orientado a aplicaciones web no transaccionales por su estructura, ya que está debidamente desarrollado para realizar consultas de tipo SELECT.

Con respecto a InnoDB¹¹, específicamente se encuentran orientados a aplicaciones transaccionales e integridad referencial, pues se encuentra desarrollado para soportar el término ACID (Atomicidad, consistencia, Integridad, Duplicidad). InnoDB permite definir restricciones de claves foranes (Foreignkey), permitiendo a los desarrolladores asegurarse que los datos con referencia a otras tablas sean validos.

Los Conectores

Son bibliotecas escrito en diferentes lenguajes de programación que permiten una conexión remota o local con servidores de MySQL con su respectiva ejecución de consultas.

¹⁰ MyISAM es el mecanismo de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL.

¹¹ InnoDB es un mecanismo de almacenamiento de datos de código abierto para la base de datos MySQL.

Gestor de Conexiones

Este gestor es el encargado de mantener las múltiples conexiones de los clientes. Se debe de configurar un gestor de conexiones para optimizar recursos como limitar el número de conexiones concurrentes, también implementa un pool de conexiones (tiempo de vida de una conexión), autenticación de usuarios.

Procesamiento y optimización de consultas

El ciclo de vida de un procesamiento consiste en la consulta previa, el análisis sintáctico, una toma de decisión (determinar el orden de lectura, escritura de las tablas y uso de índices) con el fin de que la consulta sea más eficiente.

En la optimización de consultas va a depender de una gran medida del gestor de almacenamiento, ya que este pregunta si soporta ciertas características para decidir qué tipo de optimización es el más adecuado.

Caché de Consultas

Aquí se almacena consultas y resultados enteros con el fin de evitar consultas que a veces son innecesarias por parte del procesador de consultas, para obtener resultados de consultas mucho más rápidas.

Control de Concurrencia

Es un mecanismo que permite evitar que consultas diferentes terminen en inconsistencias o efectos no deseados al consultar una misma porción de datos.

Gestión de Transacciones y Recuperación

Permite dotar de semántica, es decir, todo o nada cuando se realiza una consulta en una sola transacción. El servidor anulara cualquier proceso de ejecución mal realizada. La recuperación permite regresar a un punto anterior de una transacción.

Estructura de MySQL

La estructura de MySQL, va a depender en gran medida del tipo de motor de almacenamiento seleccionado, entre estos tenemos: MyISAM, InnoDB; como ya se menciono anteriormente difieren en algunos aspectos.

Cuando se utiliza un motor de almacenamiento InnoDB, todas las tablas de la base de datos a implementar se almacenan en un único archivo, permitiendo realizar transacciones, logrando tener claves foráneas, con lo que nos da una característica muy importante en cuando a la fiabilidad, en comparación con otros motores de almacenamiento.

InnoDB mantiene un cache de datos e índices en la memoria principal, mediante un espacio de tablas, el cual puede consistir de varios archivos o particiones de disco.

No obstante, cuando utilizamos un motor MyISAM su forma de almacenar las tablas de nuestra base de datos lo hace en tres archivos. Como se indica en la siguiente tabla 2:

Archivo	Propósito
tbl_name.frm	Archivo de formato de tabla
tbl_name.MYD	Archivo de datos
tbl_name.MYI	Archivo índice

Tabla 2: Almacenamiento de las Tablas

En la siguiente tabla se muestra como las tablas son almacenados en la Base de Datos

Un motor de almacenamiento MyISAM utiliza índices B-tree¹², los mismos que tiene varios formatos de almacenamiento, fijos y dinámicos.

Ventajas de MySQL

- MySQL es compatible con la mayoría de plataformas debido a su estabilidad general del uso de bibliotecas de subprocesos.
- Capacidad del núcleo del S.O para administrar procesos o subprocesos en diferentes CPU's, permitiendo ejecutar procesos que bloqueen y liberen procesos en una región crítica.
- Rendimiento y estabilidad general del sistema de archivos.
- Escalabilidad y límites con respecto a grandes bases de datos.
- Tiene una gran comunidad para soporte con cualquiera de nuestras dudas, problemas o inconvenientes que se puede tener al momento de desarrollar una aplicación para el cliente final.

¹² B-tree: es un algoritmo que minimiza el número de veces que un medio debe ser accesible para localizar un registro deseado, lo que acelera el proceso.

Desventajas de MySQL

- MySQL carece de procedimientos almacenados, en la parte técnica.

1.4 GENERALIDADES DE LAS BASES DE DATOS RELACIONALES

1.4.1 FUNCIONES DE UTILIZAR UNA BASE DE DATOS RELACIONAL

Al utilizar una base de datos relacional tenemos que conocer las siguientes funcionalidades:

- Garantías de ACID.
- Normalización.
- Coherencia de la información a almacenar.

1.4.2 BENEFICIOS DE UTILIZAR UNA BASE DE DATOS RELACIONALES

Contar con un RDBMS permite que la información esté disponible de forma inmediata, sea consistente, segura; los cuales nos brindan los siguientes beneficios:

- Un sistema de gestión relacional nos permite controlar la mayoría de procesos, a través de módulos.
- Permite una personalización adecuada hacia los clientes ya que proveen de herramientas para el desarrollo y adaptación de requerimientos individuales.

- Tener mejores estrategias de negocios que conlleven a una mejor relación con el consumidor final.
- Crecimiento de un negocio, elevando la rentabilidad de la inversión.
- Disponibilidad de información desde cualquier parte del mundo, si la aplicación se encuentra orientado en la web.

1.4.3 VENTAJAS DE UNA BASE DE DATOS RELACIONAL

A continuación mencionaremos algunas ventajas de las bases de datos relacionales:

- No se tienen la necesidad de utilizar el papel como fuente de almacenamiento primario.
- La actualización de la información en una base de datos es mucha más rápida.
- Disposición de información precisa y actualizada en momentos justos.
- Reducción de información repetida durante el ingreso de datos.

1.4.4 DESVENTAJAS DE UNA BASE DE DATOS RELACIONAL

En lo referente a las desventajas de las bases de datos relacionales, se debe de tener en cuenta que lo más complejo de entender es la manera de cómo se relacionan las diferentes entidades de la base de datos.

CAPÍTULO 2

DESVENTAJAS DEL USO DE BASE DE DATOS RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES

2.1 ANÁLISIS DE LAS BASES DE DATOS GNU

La palabra GNU tiene referencia a software libre, es decir, el usuario puede ejecutar, copiar, distribuir, estudiar, modificar o mejorar el software con libertad¹³; por tanto se puede tomar este término para referirnos a que existen bases de datos libres sin ningún tipo de restricción de precio o licencias.

La realización de proyectos de código abierto funciona eficientemente cuando, al utilizar una base de datos libre su aplicación se lo realiza con herramientas básicas y donde cada uno de sus requerimientos es definido correctamente.

Mientras que las bases de datos propietarias, el código fuente solo es desarrollado y conocido por un número reducido de personas; en las bases de datos de código abierto el interés es más global, donde se puede aportar ideas disponiendo de los conocimientos necesarios.

En los proyectos de código abierto, el rendimiento y la prueba de su funcionalidad deben de tener un enfoque muy estructurado, así también, como los requerimientos de licencias, en el empaquetado (packaging), actualizaciones y mejoras.

¹³ <http://www.gnu.org/philosophy/free-sw.es.html>

La ventaja monetaria del software de código abierto es que no tiene costo de su licencia, pudiendo el usuario acceder al código fuente, brindando independencia del proveedor, la información es abierta, no existe una política de ocultamiento, en el caso que exista errores en el funcionamiento, el usuario tiene la libertad de usar para cualquier propósito, modificando y adaptando según sus necesidades, distribuyendo o haciendo alguna mejora en beneficio de la sociedad.

2.2 ANÁLISIS DE BASE DE DATOS COMERCIALES

Una base de datos comercial se puede entender aquella que, para su utilización se encuentra relacionada con las licencias que se debe de adquirir, significando costos altos, dependiendo de la versión o de las diferentes características que posea dicha versión.

Las inversiones que realizan estas bases de datos en sus sistemas son bastantes notorias, pues al hablar de una base de datos comercial, se puede notar que existen diversas herramientas que pueden ayudar a administrar de una mejor manera los datos, herramientas como: para probar rendimiento, herramienta para la monitorización en tiempo real, en herramientas de ajustes de aplicación, herramientas de diagnósticos de falla, entre otras herramientas.

Las diferentes herramientas que se presentan en las bases de datos comerciales serán de gran ayuda hoy en día para los administradores, ya que constantemente nos encontramos en entornos de datos con cambios frecuentes y estas herramientas ayudan al administrador a mejorar el tiempo de respuesta para responder en forma más eficiente las peticiones o consultas que se generan periódicamente.

No obstante, no debemos olvidar del soporte que ofrecen, en muchas de los casos pueden tener soporte las 24 horas del día, logrando tener una ventaja de costo vs beneficio. Al tener esta ventaja, vamos a tener la seguridad de que si, se produjera algún tipo de fallo, nuestro soporte nos dirá que hacer.

Al ser bases de datos que incluyen servicios de soporte y administración adicional, (soporte que brindan sobre sus transacciones, su estabilidad, escalabilidad, multiplataforma) las empresas u organizaciones pueden tomar la decisión de utilizarlas.

Otro aspecto también importante es la seguridad de la plataforma y las políticas en el suministro de los parches de seguridad y actualizaciones que han sido modificadas para el incremento en el nivel de usuarios.

Se tiende a mejorar servicios en copias de seguridad y recuperación ante fallos ya que funcionan de modo que se minimiza en lo posible el trabajo perdido cuando se produce un fallo.

En síntesis, una base de datos comercial hoy en día presentan muchos desafío, entre los cuales se describen las siguientes:

- El rendimiento: Consiste en cómo mantener a un punto máximo de rendimiento los servicios comprometidos.
- Administración de cambios: Hace referencia al cómo reducir el riesgo de implementación de cambios a costos bajos, como innovaciones en tecnología.

- Administración actual: Se trata de automatizar las diferentes tareas que se procesan diariamente, con el objetivo de acoger planes más estratégicos, como seguridad y una alta disponibilidad.¹⁴

2.3 DESVENTAJAS

Al realizar el análisis entre las bases de datos comerciales y libres, podemos mencionar algunas desventajas que presentan cada una, mencionando a continuación algunas de estas:

Con referente a los precios, las bases de datos comerciales son relativamente costosas, a comparación de las bases de datos libres; aunque va a depender de los requerimientos que lo entidad u organización lo solicite, además de las versiones del producto.

Las configuraciones mal realizadas en una base de datos comercial pueden traer repercusiones muy elevadas, haciendo que la base de datos en su rendimiento disminuya notablemente.

En la mayoría de los casos, las bases de datos comerciales requieren de una enorme cantidad de recursos para su instalación, entre estos recursos podemos mencionar la memoria RAM, servidores de alto rendimiento.

Existen bases de datos comerciales que se restringen en la usabilidad entre los diversos sistemas operativos que existen actualmente, instalando en un solo sistema operativo.

¹⁴ Oracle Database 11g Release 2: Manageability and Real Application Testing Overview

A continuación se describen los puntos en contra de las bases de datos libres:

Existen herramientas adicionales en diversas bases de datos libres, pero su adquisición tiene un costo extra por alguna restricción por parte del proveedor, entre estas se puede mencionar una suscripción anual, trimestral.

Además, se requiere de personal altamente calificado, que tenga conocimientos sólidos para estas tecnologías en bases de datos.

El problema principal de las bases de datos de código abierto es por el soporte técnico, ya que al existir un problema de configuración no tendremos una ayuda por parte de un asesor comercial.

CAPÍTULO 3

ESTUDIO DE BASE DE DATOS NO RELACIONALES

3.1 INTRODUCCIÓN

En la actualidad las aplicaciones web nos permiten acceder a sistemas desarrollados en diversos lenguajes de programación, para mantener información en una variedad de sistemas de bases de datos.

Los sistemas informáticos desarrollados, no siempre cubren las expectativas y requerimientos de los usuarios. Esto puede ser debido a problemas en el diseño y desarrollo del sistema (no incluir a los usuarios en el diseño, diseño inadecuado, problemas de configuración, cambios en los requerimientos cuando se está en una fase avanzada del desarrollo, falta de conocimientos de la tecnología), teniendo en cuenta todos estos puntos podemos alcanzar un desarrollo óptimo, con una preparación adecuada tanto en su análisis y la estructura de su diseño.

¹⁵Por otro lado está la amplitud existente en el mercado, en donde nuestras aplicaciones deben satisfacer las necesidades de los usuarios, adaptándose en eventos nuevos que se pueden dar como una aplicación multilenguaje, los accesos a través de nuevos y distintos dispositivos; una de las formas es desarrollar aplicaciones en donde cada una de sus partes puedan ser modificadas independientemente unas de otras, obteniendo así que las partes que han sido modificadas no afecte a todas las partes, para esto se suelen utilizar desarrollos divididos comúnmente llamados capas para minimizar así la dependencia entre cada una de estas, convirtiendo así en un problema de mantenimiento.

¹⁵ Antonio on 16 noviembre, 2011 · leave a comment · in base de datos, nosql

¿Qué son las Bases de Datos No Relacionales?

La mayoría de las aplicaciones son soportadas por bases de datos relacionales, pero con el transcurso de los años se han ido mejorando satisfactoriamente con una normalización adecuada, dependiendo de la demanda en su entorno y utilizando sistemas de persistencia.

Con la evolución de Internet, y los servicios disponibles, desde hace aproximadamente una década, cada vez se ha dado mayor espacio a que los usuarios interaccionen con los sistemas como la web 2.0. Esto ha propiciado que servicios como redes sociales (Facebook, Twitter, Orkut, por ejemplo), blogs, etc., hayan consolidado y popularizado a nivel mundial, generando así los problemas que existen de escalabilidad y rendimientos de las bases de datos relacionales en el manejo de información no transaccional, de aquí la aparición de las bases de datos no relacionales siendo estas capaces de dar una solución a los problemas existentes en las bases de datos relacionales.

Las bases de datos no relacionales no es solo una de base de datos sino una solución para almacenar grandes cantidades de datos, son sistemas de acaparamiento de información se almacenan de una manera diferente donde sus columnas son dinámicas, pudiendo realizar cambios sin perder la agrupación de la información, no aplican una estructura de datos en forma de tablas y las relaciones entre ellas, volviéndose de esta manera más flexibles y permitiendo almacenar la información en otros formatos como: clave-valor, Mapeo de Columnas, Documentos o Grafos.

La principal característica de las bases de datos no relacionales es que nos permiten manipular grandes cantidades de información de una manera muy rápida (su almacenamiento se realiza en memoria) y tienen la capacidad para escalar horizontalmente sin perder rendimiento, funcionando de una manera óptima con hardware de bajo coste, pudiendo añadir nuevas máquinas en caliente sin necesidad de reiniciar el sistema.

El modo de almacenamiento de la información de estas bases de datos es de una manera distinta al que normalmente se realiza por tablas, generalmente lo hacen como clave – valor; esto simula una tabla en la que las columnas son dinámicas, pueden darse cambios sin que se pierda la agrupación de la información teniendo así clases con varios atributos y otras no, de esta manera se pueden dar cambios en la estructura de la información sin tener que rediseñar todo nuevamente.

Las bases de datos relacionales al realizar una consulta, internamente se realizan diversos pasos como: convertir, preparar, optimizar, procesar, leer el disco y ejecutar de esta manera obtenemos un rendimiento demasíadamente elevado sobre todo en grandes cantidades de datos, sin embargo en las bases de datos no relacionales el tiempo de ejecución en una consulta evita hacer todos estos pasos accediendo de la memoria lo que nos interesa, permitiéndonos que cada una de las consultas sean rápidas y eficientes.

3.2 CONCEPTOS GENERALES

3.2.1 DEFINICIÓN DE NoSQL

NoSQL es un sistema de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales (RDBMS). Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, coherencia, aislamiento y durabilidad), y habitualmente escalan bien horizontalmente¹⁶.

¿Quién utiliza las Bases de Datos No Transaccionales?

No son utilizadas para todas sus aplicaciones, solo para algunas partes de sus sistemas empresariales, estas son:

¹⁶ Wikipedia – 12 de abril del 2012.- http://es.wikipedia.org/wiki/Modelo_relacional

Las bases de datos NoSQL comenzó con las principales compañías de Internet, como Google, Amazon, Twitter y Facebook, ya que estas tenían que enfrentarse a desafíos. Con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información de grandes volúmenes de datos que tenían unas estructuras horizontales más o menos similares, dándose cuenta que el rendimiento y sus propiedades de tiempo real eran más importantes que la coherencia, en la que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

3.2.2 OTROS TÉRMINOS NoSQL

Entre otros términos que nos pueden ser de guía en este estudio tenemos los siguientes:

- NoSQL literalmente es una combinación de dos palabras. No y SQL.
- Los creadores y pioneros del término NoSQL en un principio quisieron decir No RDBMS or No relacional pero optaron por el sonido de NoSQL y adoptaron este término.
- Algunos propusieron NonRel como alternativa a NoSQL.
- Otros propusieron al termino NoSQL como una palabra que expanda a “Not Only Sql”.
- Pero, hoy en día la palabra NoSQL en términos generales es usado para todas la bases de datos y almacenes de datos que no siguen los populares y bien establecidos principios RDBMS¹⁷.

3.3 TIPOS DE BASES DE DATOS NO RELACIONALES

¹⁷ PROFESSIONAL NOSQL – SHASHANK TIWARIN 2011

Existen varios tipos de bases de datos no relacionales; a continuación describimos cada una de ellas dependiendo del almacenamiento de la información:

3.3.1 ALMACENES KEY - VALUE (CLAVE - VALOR)

Cada uno de los elementos se identifica por una llave por medio del cual podemos recuperar la información de forma rápida, este valor por lo general es almacenado como un objeto BLOB (nombre que se da a datos que son objetos binarios de gran tamaño ej. Imágenes, música, etc.), ya que para la base de datos lo importante no es su contenido sino la clave y el valor.

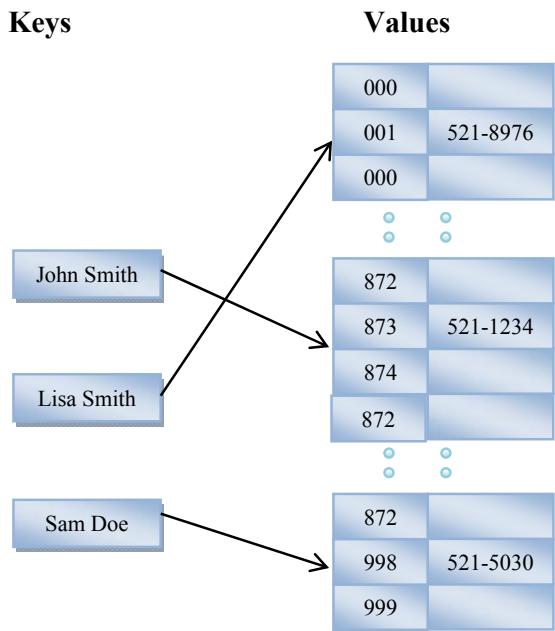


Figura 6: Almacenes Key – Value

Una característica importante es que se puede escalar fácilmente haciendo particiones de los valores de acuerdo a su clave, si una clave está entre 1 y 1000 va a un server, los de 1001 a 2000 a otro y así sucesivamente. Dentro de estas bases de datos podemos encontrar:

✓ **BigTable de Google**

Su almacenamiento físico se realiza en GFS (Google File System), el cual es un sistema de archivos distribuido propietario de Google, las bases de datos tradicionales no permiten la creación de sistemas lo suficientemente grandes y la posibilidad de que puedan ser distribuidas en miles de servidores. Por lo cual la creación de BigTable en donde la información a almacenar es grande en orden de Petabytes, cada una de las tablas pueden ocupar 200 Megabytes, la información para ser enviada se comprime a través de un sistema de compresión propietario de Google. Cuenta con una interfaz API de uso público permitiendo que cualquier desarrollador trabaje sobre el mismo.

Al ser un mapa distribuido ordenado de manera multidimensional, se puede añadir una tercera dimensión, estas son: filas, columnas y tiempo.

Las claves de las filas son cadenas con un tamaño parcial, su operación es atómica (no se realizan dos operaciones en la misma fila), las filas son particionadas en tablets dispersándose en varios nodos servidores.

Al ser Google propietario no existe la posibilidad de emplearlo ni hacer pruebas sobre el mismo.¹⁸

✓ **SimpleDB de Amazon**

Es un almacén de datos no relacionales de alta disponibilidad y flexible.

¹⁸ <http://www.nosql.es/blog/nosql/google-bigtable.html>

Se puede centrar en el desarrollo de aplicaciones sin que existan una preocupación de las infraestructuras, la disponibilidad, mantenimiento del software, la gestión de esquemas e índices.

Ofrece una interfaz de servicios Web simple para crear, almacenar y consultar fácilmente los datos y obtener los resultados de las consultas de una manera correcta.

Los datos se indexan automáticamente, de forma que es fácil y rápido encontrar la información necesaria.

No es necesario predefinir ni cambiar esquemas si se añaden datos posteriormente.

✓ **Cassandra**

Es una base de datos de código abierto. Fue iniciada por Facebook para probar la solvencia de problemas relacionados con el rendimiento del motor de las búsquedas, es decir la comunicación entre los usuarios, implicando un gran volumen de los datos a almacenar con un crecimiento alto y la necesidad de ofrecer un nivel de calidad excelente.

Fue diseñado para que las configuraciones de explotación fueran escalables, horizontales y económicas.¹⁹

¹⁹ <http://www.nosql.es/blog/nosql/cassandra.html>

✓ Riak

Es una base NoSQL key-value de código abierto, escalable y simplifica el desarrollo, dando a los usuarios la capacidad de formar rápidamente prototipos, probar y desplegar sus aplicaciones.

Mencionamos algunas características de Riak:

Escalabilidad

Riak distribuye automáticamente los datos de todo el grupo (equipos disponibles) y se obtiene un aumento de rendimiento casi lineal a medida que agrega capacidad de recursos de almacenamiento.

Tolerancia a Fallos

El clúster soporta los fallos en los nodos o en sus partes, no se pierde datos ante estos fallos, el sistema es diseñado con fallos de máquinas e interrupciones de servicio de red.

Alta disponibilidad

El sistema debe estar disponible tanto para las lecturas y escrituras. La arquitectura de share - nothing (nada compartido), haciendo que cada nodo en el clúster este habilitado para realizar lecturas y escrituras, siendo la disponibilidad garantizada.

Replicación

Dependiendo las necesidades, a riak se le puede configurar para replicar entre múltiples clústeres iguales o diseñar uno o más clústeres de lectura o backup. Cada

nodo en el clúster responde a las peticiones de lectura - escritura y el evento de replicación asegura que todos los clústeres sean actualizados.

Cada uno de los clústeres que se configura en Riak, consta de los siguientes elementos:

- **Nodos:** nodos físicos, o las máquinas de las que está compuesta el clúster.
- **Nodos Virtuales:** nodos virtuales que corren en cada máquina. Cada máquina física puede contener uno o varios nodos virtuales.
- **Particiones:** cada clúster Riak tiene un entero de 160bits dividido en particiones.

Voldemort

Es un sistema distribuido de bases de datos clave - valor. Aplica el concepto clásico de clúster, cada uno de sus nodos son independientes (no maestro - esclavo).

Es utilizado por empresas que requieren procesar grandes cantidades de información con alto número de operaciones concurrentes y su tiempo de respuesta rápido.

MemcacheDB

Es un sistema de almacenamiento clave-valor distribuido y diseñado para persistencia.

No se trata de una solución de caché, pero si sigue el control memcached, el mismo que es sistema distribuido donde todas sus operaciones son realizadas en caché de memoria, por lo que cualquier cliente de memcached puede conectar con él.

Utiliza como sistema base de almacenamiento Berkeley DB (una de las más rápidas), por lo que muchas de sus características incluye transacciones y replicación. Tiene un alto rendimiento de lectura/escritura para objetos basados en claves-valor. Entre los comandos de memcache soporta get, set, add, replace, append, prepend, incr, decr, delete y stats. MemcacheDB

3.3.2 ORIENTADAS A COLUMNAS

Esta base de datos guarda cada uno de sus valores en columnas y no en filas, como se puede ver en la figura 8, con este cambio podemos ganar más velocidad al momento de realizar las lecturas y mucho más si la consulta es un número reducido de columnas, pero no es eficiente para realizar escrituras es por esto que este tipo de soluciones se usa en aplicaciones con un número bajo de escrituras.

Title	Title	Title	Title	Title
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data

Tabla 3: Base de Datos Orientada a Columnas

3.3.3 BASADA EN DOCUMENTOS

Estas bases de datos almacenan su información como un documento con una estructura simple como JSON o XML con una llave única, siendo similar a las bases de datos Key – Value con la diferencia que su valor es un archivo, cuando el servidor entiende cada uno de los datos este puede realizar las operaciones sobre ellos.

Las implementaciones con esta base de datos permiten que las consultas sean muy avanzadas, estableciendo las relaciones entre ellos, la única desventaja es que no permiten realizar JOINS (combinar registros de dos o más tablas), encontramos entre las más importantes a MongoDB y CouchDB.

En la tabla 3, se describe un ejemplo de la manera de cómo se almacenan los datos en una base no relacional las cuales se utilizan con un formato JSON es decir no dependen de una estructura rígida.

Clave	Documento
10	Nombre: Mateo; Apellidos: Dávila; Nacionalidad: Ecuatoriana
11	Nombre: Ángeles; Apellidos: Córdova ; Nacionalidad: Ecuatoriana; Edad: 30;
12	Nombre: Fernanda; Nacionalidad: Ecuatoriana

Tabla 4: Base de Datos Orientada a Objetos

En la tabla 4, observamos un ejemplo de una tabla en donde sus datos se almacenas bajo una estructura de datos definida.

Clave	Nombre	Apellidos	Nacionalidad	Edad
63	Pepe	García	Española	
64	Francisco	Cevallos	Francesa	29
65	Mario		Italiano	

Tabla 5: Base de Datos Relacional (estructura de los datos)

Podemos ver en la tabla 4, que en la NoSQL los campos vacíos no se añaden y que se pueden añadir campos concretos a documentos concretos, sin tener que aumentar el número de columnas.

Este tipo de bases de datos son simplemente un almacén key-value con la excepción de que el valor no se guarda sólo como un campo binario, sino con un formato definido de forma tal que el servidor pueda entenderlo. Esto no quiere decir que tengan un esquema, siguen siendo schema-less, seguimos teniendo solo 2 campos y uno de ellos binario.

La diferencia es que el campo binario puede ser entendido por la base de datos. Dicho formato a menudo es JSON,²⁰ pero puede ser XML o cualquier otra cosa. Si el servidor entiende los datos, puede hacer operaciones con ellos. De hecho varias de las implementaciones de este tipo de bases de datos permiten consultas muy avanzadas sobre los datos, e incluso establecer relaciones entre ellos, aunque siguen sin permitir joins por cuestiones de rendimiento. Todo esto sin perder de vista que sigue siendo un key-value con todas las ventajas que nos dan estas bases.

²⁰ JSON: es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Por ello este tipo de bases de datos son en gran medida los responsables del auge actual de NoSQL. Ofrecen una gran performance y escalabilidad sin perder del todo los beneficios del modelo relacional. Entre las bases de datos de este tipo, están los más famosos ejemplos de la familia NoSQL: MongoDB y Apache CouchDB.

3.3.4 ORIENTADAS A GRAFOS

La información que almacena esta base de datos lo hace en forma de grafo, dando la importancia no solo a los datos sino también a cada una de las relaciones que existen, estas relaciones pueden tener atributos y realizar consultas directas a las mismas y al estar almacenadas de esta manera es más eficiente y fácil de navegar.

Este tipo de bases de datos son aprovechadas si la información es representada fácilmente como en las redes sociales o sistemas que recomienden algún producto, la ventaja que tienen estas bases de datos es que se pueden aplicar algoritmos estadísticos para recomendaciones y tienen que recorrer cada uno de los grafos.

Entre las implementaciones más usadas está Neo4J, Hyperbase- DB e InfoGrid.

En la tabla 5, se presenta un resumen de las características de los distintos tipos de bases de datos no relacionales.

Tipo	Características
Almacenes (Clave - Valor)	Key-Value <u>Cassandra</u> , <u>Membase</u> , <u>Redis</u> o <u>Riak</u> . Almacenan los datos en pares clave-valor (un valor podría ser un objeto).
Orientadas a Columnas	Vertica, Microsoft, Qd Technology, Sybase, ParAccel, Infobright. Mejor rendimiento. Eficiente en el almacenamiento. Cambios en el esquema tiene menor impacto y por lo tanto el coste de realizarlo es menor.
Basada en Documentos	<u>MongoDB</u> y <u>CouchDB</u> . Almacenan los datos en formatos estructurados (registros) como JSON.
Orientadas a Grafos	<u>Neo4J</u> y <u>FlockDB</u> . Almacenan los elementos y las relaciones entre ellos con un estilo de grafo (para redes sociales, redes de transporte, mapas de carreteras, topologías de red).

Tabla 6: Características de las Bases de Datos

3.4 GENERALIDADES DE LAS BASES DE DATOS NO RELACIONALES

3.4.1 FUNCIONES

En nuestro mundo actual, donde la información es primordial en cada una de las etapas, almacenamiento, recuperación y visualización; se nos presenta una

necesidad muy grande, que consiste en el procesamiento de la información de una forma eficaz y eficiente.

Una base de datos no relacional es una alternativa que se nos presenta actualmente para este procesamiento de la información donde consiste en guardar la información en un almacenamiento de datos mucho más rápido, sin ningún tipo de estándar convencional, y sobre todo escalable.

3.4.2 BENEFICIOS

Al enfocarnos en bases de datos no relacionales, una pregunta primordial sería: el porqué del cambio de que muchas empresas reconocidas en el mundo lo están haciendo, migrar sus bases de datos tradicionales (relacionales) a un almacenamiento de datos en la mayoría libres, sin llegar a tener una estructura convencional.

Entre las respuestas para este tipo de almacenamiento tenemos:

- En la velocidad en el tiempo de respuesta, ya que los usuarios necesitan que sus necesidades sean resueltas en el menor tiempo posible.
- La cantidad de información que se puede almacenar en un instante dado.

3.4.3 VENTAJAS DE LAS BASES DE DATOS NO RELACIONALES

- Hay bases de datos no relacionados de código abierto
Los productos de código abierto aportan beneficios a los desarrolladores como: en precio, no se necesita de un servidor con grandes recursos, no

tienen una estructura de datos definida, los datos pueden ser diversos existiendo heterogeneidad.

➤ **Es de escalamiento sencillo**

Las bases de datos no relacionales buscan una manera de añadir más servidores para manejar más cargas de datos, permitiendo a las empresas una distribución de los equipos dependiendo de las actividades a realizar.

➤ **Economía**

Las bases de datos no relacionales utilizan servidores de bajo costo para la administración de los datos y el volumen de las transacciones que realicen. El costo por gigabyte por segundo para estas bases puede ser mucho menor que el costo de los RDBMS, lo que le permite almacenar y procesar más datos a un precio mucho más bajo, pudiendo así añadir máquinas según sean las necesidades de la empresa.

➤ **No generan cuellos de botella**

Las bases de datos relacionales tienen este problema ya que estos tienen que transcribir cada sentencia para ser ejecutadas incluyendo las sentencias complejas, además de un nivel de ejecución más preciso para llevar a cabo, por lo que constituye un punto de entrada común, único y conflictivo en base a rendimiento.

3.4.4 DESVENTAJAS DE LAS BASES DE DATOS NO RELACIONALES

➤ **Soporte**

Las empresas que utilicen estas bases de datos buscan que si su sistema falle por algún motivo tengan el soporte necesario, oportuno y competente de parte de sus vendedores, que en este caso no es el adecuado.

➤ **Formalismo**

A pesar de que en algunas empresas ya se ha utilizado estas bases, todavía se enfrentan a problemas de credibilidad e inestabilidad.

➤ **Experiencia**

El corto tiempo de existencia de estas bases de datos implica que no haya un número considerable de desarrolladores y de administradores que conozcan de esta tecnología lo que es un problema encontrar personas con conocimientos técnicos lo suficientemente necesarios para que las empresas se centren en esta base de datos para sus aplicaciones.

➤ **Administración**

Los objetivos de esta base de datos es proporcionar una solución de administración desde cero pero es un problema que esto se llegue a dar ya que requiere de mucha habilidad de instalar y de un gran esfuerzo para mantener.

➤ **Compatibilidad**

Estas bases de datos tienen su propio API, las interfaces de consultas son únicas y tienen peculiaridades.

3.4.5 ESTRUCTURA DE LA BASE DE DATOS NO RELACIONAL A IMPLEMENTAR

Elegir una Base Datos no Relacional (NoSQL) para el desarrollo de una aplicación prototipo nos resulta complicado, ya que tenemos varias alternativas cada una con sus funciones y reglas establecidas, debemos tener en cuenta de las necesidades de las empresas, las cuales van a depender del área a donde se vayan aplicar.

Al realizar un estudio de la aplicación prototipo a realizar para un sitio web de blogs y la más acorde para este caso tenemos MongoDB que pertenece a las bases de datos orientadas a documentos.

En base al análisis realizado en el capítulo 3, entre las bases de datos no relacionales revisadas, hemos elegido a MongoDB por las siguientes razones:

La razón primordial que se escogió MongoDB como nuestra base de datos a analizar fue por su similitud con las bases de datos relacionales, teniendo en cuenta que una colección es semejante a una tabla, un documento es similar a una fila y un campo a una columna.

Además MongoDB ofrece en su sitio web un gran número de drivers nativos oficiales para diferentes lenguajes de programación.

Y con respecto a su almacenamiento y modelo de datos se utiliza un formato JSON, la cual estructura los datos sin perder el entendimiento de los mismos.

A continuación se describe algunas características que podemos destacar de esta base de datos:

- Es una base de datos no relacional de código abierto.
- Está orientado a la creación de wikis, blogs, y sistemas de gestión documental.
- No soporta Joins como las bases de datos relacionales.

- Son libres de esquemas.

MongoDB

Es un almacén de documentos, donde cada uno de los documentos son agrupados dentro de colecciones o en una sola colección, cada una de estas colecciones pueden ser comparadas como una tabla relacional, la gran diferencia que no se impone a una estricta restricción de esquema como lo hacen en las tablas relacionales.

Las colecciones pueden ser almacenadas en un lugar denominado namespaces pero su representación no es tan jerárquica; además una colección está formada por una o más documentos.

Cada documento es almacenado en un formato BSON, tiene un único identificador y está formado por uno o más campos.

MongoDB se puede comparar con las una base de datos relacional por los siguientes términos según la siguiente tabla:

RDB	MongoDB
Tabla	Colección
Fila	Documento
Columna	Campo o atributo

Tabla 7: MongoDB vs Base de Datos Relacional

En la siguiente tabla se muestra una comparación de una base relacional con MongoDB verificando lo que cada base contiene en el momento de su implementación

Esquemas de Modelos de Datos

En este apartado, se presenta la comparación una estructura específica de una base de datos modelada en forma relacional y no relacional.

MongoDB utiliza una colección como el principal término, a continuación se presenta una comparación sobre un esquema relacional y un esquema no relacional, con el fin de entender la estructura de almacenamiento que se tiene en ambos casos:

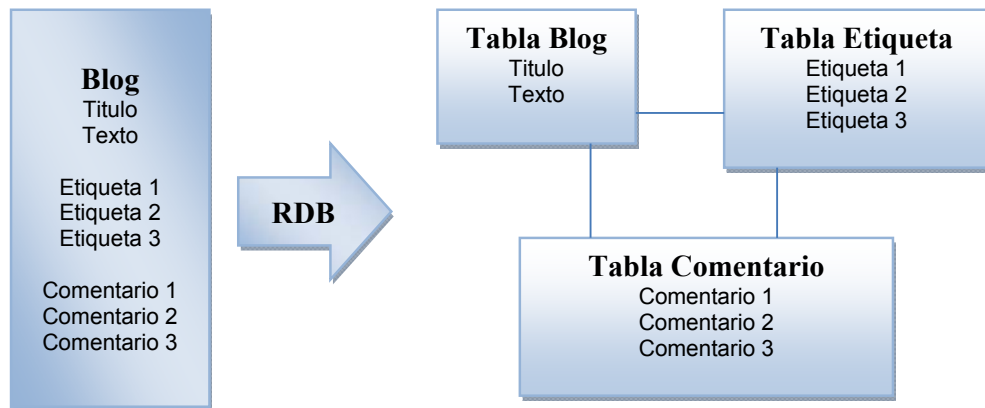


Figura 7: Esquema de una Base de Datos Relacional

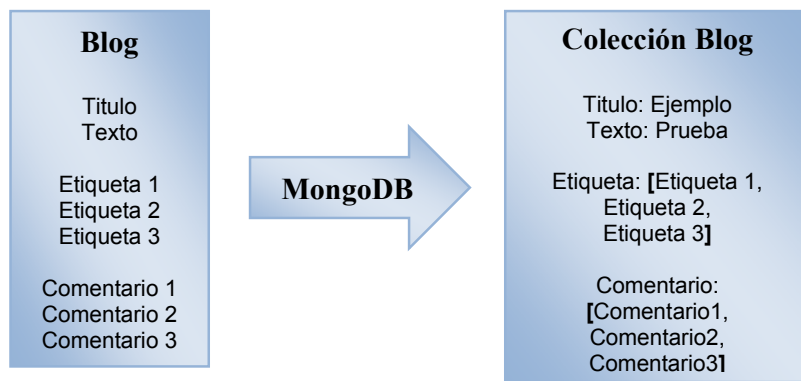


Figura 8: Esquema de MongoDB

Almacenamiento de datos en archivos asignados en memoria

Un archivo asignado en memoria es un segmento de memoria virtual, esto implica que las aplicaciones interactúan con tales archivos como si fueran parte de la memoria principal.

Manipular la memoria es mucho más rápido que realizar llamadas de sistema. La manipulación de región asignada de memoria es comúnmente llamada cache de página, la misma que ya es implementada en algunos kernel de sistemas operativos.

La estrategia de utilizar archivos memoria asignada se realiza de una manera inteligente a continuación describimos los pasos necesarios:

- Primero se evita que exista una separación entre el cache de memoria del sistema operativo y la cache de la base de datos; por ello no hay redundancia de cache.
- Segundo el caching es controlado por el sistema operativo porque la asignación de memoria virtual no es igual en todos los sistemas operativos.
- Tercer se puede ampliar la memoria caché de base de datos a utilizar toda la memoria disponible sin ninguna configuración adicional.

Cuando se asigna memoria se tiene pocas limitaciones, al implementar una base de datos MongoDB en un sistema operativo de 32 bits la capacidad máxima de datos es de 2GB, mientras en un sistema operativo de 64 bits no se aplica estas restricciones.

El tamaño de la base de datos no es la única limitación de tamaño, sin embargo existe otra limitación que rige en el tamaño de cada documento y el número de la colección de un servidor MongoDB que puede contener. Un documento no puede ser más grande a 8MB, lo que significa usar MongoDB para almacenar en

BLOBS no es muy apropiado; sino que existe GridFS para almacenar documentos de más de 8MB.

Además existe un límite en el número de namespace que pueden ser asignados en una instancia de base de datos, el número por defecto de namespace admitidos es de 24.000; asignando a cada colección un namespace significando por defecto dos índices por colección permitiendo un máximo de 8.000 colecciones por base de datos, generalmente. Sin embargo, si es necesario se puede aumentar el tamaño del espacio de nombres más allá de 24.000.

Aumentar el tamaño del namespace tiene implicaciones y limitaciones; cada namespace de colección utiliza hasta unos pocos kilobytes.

En MongoDB, un índice es implementado como un árbol-B, cada página de este árbol-B es de 8 KB, por lo tanto, la adición de namespace adicionales, ya sea para colecciones o índices, implica la adición de unos pocos KB para cada instancia adicional. Los namespace para una base de datos MongoDB se llaman **mydb** y se mantienen en un archivo llamado mydb.ns. Un archivo .ns como mydb.ns puede crecer hasta un tamaño máximo de 2GB.

Escalamiento Horizontal

Una razón común para el uso de MongoDB está en su esquema de colecciones y el otro en su capacidad de un buen desempeño a gran escala. MongoDB soporta auto - sharding para escalar horizontalmente con facilidad.

El concepto fundamental de sharding es bastante similar a la idea de columnas de las bases de datos principales donde los datos son distribuidos en múltiples servidores.

MongoDB permite ordenar colecciones para ser guardadas a través de múltiples máquinas; por lo tanto cada máquina guarda una parte de una colección es un fragmento, estos fragmentos son replicados para permitir conmutación de error. Una colección larga podría ser dividida en 4 fragmentos y cada fragmento disponible puede ser replicado tres veces. Esto crearía 12 unidades de un servidor MongoDB, las dos copias adicionales de cada fragmento de servidor es una unidad de conmutación de error.

Los fragmentos son niveles de colección y no son niveles de base de datos. Así una colección en una base de datos puede permanecer en un nodo, Donde como otro en la misma base datos puede ser compartida en múltiples nodos. Cada fragmento de almacén continuo se almacena en documentos ordenados. Cada paquete es llamado como chunks en MongoDB, Cada chunk es identificado por 3 atributos, la primera llave de documentos como llave mínima, la segunda llave de documentos como llave máxima, y por último la colección.

CAPÍTULO 4

ANÁLISIS DE BASES DE DATOS NO RELACIONALES Y RELACIONALES EN APLICACIONES WEB NO TRANSACCIONALES.

4.1 IMPLEMENTACIÓN DE UNA APLICACIÓN WEB

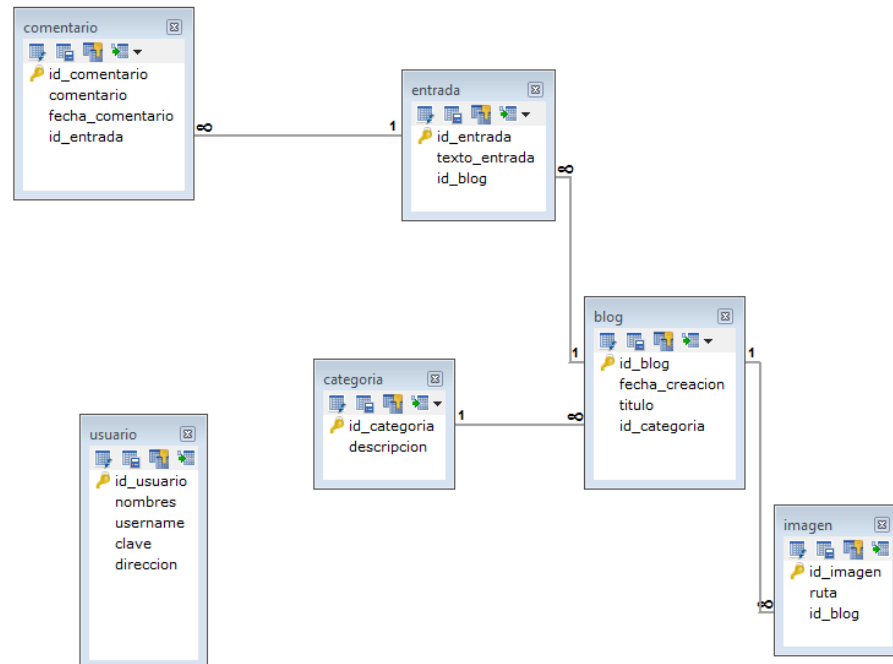
Se realizó un análisis comparativo entre la funcionalidad de la Base de Datos no Relacional MongoDB con una Base de Datos Relacional MySQL, para lo cual, se ha realizado un prototipo que permite manejar información de un blog, que tiene las siguientes funcionalidades:

- Permite visualizar las categorías que existen.
- Permite realizar consultas de blogs a través de palabras.
- Permite realizar consultas de blogs, a través de subcategorías que se visualizan.
- Muestra los resultados según el título, categoría, imágenes si existen en el blog, entradas y sus comentarios con la fecha que fueron ingresados.

Aplicación Web con una Base de Datos Relacional

Para realizar nuestra aplicación web, se realizaron los siguientes pasos:

1. Se realizó un diagrama de entidad relación de un blog, como se muestra a continuación:



2. Se instaló un servidor independiente de plataforma, de software libre XAMPP donde contiene un servidor web (Apache), una base de datos(MySQL) y un intérprete para lenguaje de script.(PHP), en la siguiente tabla se indica las versiones de los software que se utilizaron:

Software	Versión
XAMPP	1.7.7
APACHE	2.2.21
PHP	5.3.8
MYSQL	5.5.16

Tabla 8: Versiones de Software

En la tabla se muestra las versiones del software utilizados para la aplicación con Base de Datos Relacional

3. Se instaló una herramienta de interfaz gráfica para la administración de la base de datos MySQL.

4. Para la carga de una base de datos de gran volumen se tuvo que realizar un programa, el mismo que consistió en leer un archivo de texto de gran tamaño, parara luego substraer los caracteres según sea el caso y posteriormente insertando en nuestra base de datos MySQL, a continuación se puede ver el método que se utilizo:

```
public Connection ingresoDatosBlog(int longitud, int lon_cat)
{
    Connection miConexion;
    miConexion=ConexionDB.GetConnection();
    String valfechaBlog;
    int valIdCat=0, valIdUsr=0;
    try{
        Statement st=(Statement)miConexion.createStatement();
        FileInputStream fstream = new
        FileInputStream("B:/PERSONAL/BERNARDO/TESIS/final.txt");
        DataInputStream entrada = new DataInputStream(fstream);
        BufferedReader buffer = new BufferedReader(new InputStreamReader(entrada));
        String strLinea, cadena, resultado= "";
        char letras;    String var="";    int sec=86990;
        while ((strLinea = buffer.readLine()) != null){
            for (int i=0;i<strLinea.length();i++) {
                letras=strLinea.charAt(i);
                Character c=new Character(letras);
                cadena= c.toString();
                resultado=resultado.concat(cadena);
                if (resultado.length()==longitud || resultado.length()>=longitud ){
                    valfechaBlog=generarFechaBlog(33, 1, 12, 1, 28, 1);
                    var=resultado;
                    valIdCat=((int)(Math.random()*lon_cat+1));
                    st.execute("insert into blog values
                    (""+sec+"",""+valfechaBlog+"",""+resultado+"",""+valIdCat+"") /*,""+valIdUsr+"")");
                    resultado="";
                }
                else if (resultado.length()<longitud){continue;}sec++;}
            entrada.close();
            st.close();
            miConexion.close();
        }
        catch (Exception e){ //Catch de excepciones
            System.err.println("Ocurrio un error: " + e.getMessage());
        }
        return miConexion;
    }
```

5. A continuación, se procedió a la creación de nuestro proyecto, basándonos en los paquetes antes descritos, donde podemos ver en el [anexo 1](#).

6. Para comprobar el uso de nuestro proyecto en estos entornos se realizó la siguiente prueba:

1. Prueba de Carga ([Ver Resultados](#))

Aplicación Web con una Base de Datos No Relacional

1. Se instaló la base de datos MongoDB, la misma que se siguieron los siguientes pasos:

- Se crea una carpeta en el disco duro, donde se almacenara los archivos de registros de sucesos que se realice (logs).
- Se crea otra carpeta en el disco duro, donde se almacenara la base de datos (data).
- Luego en una consola de sistema, se ejecutarán los siguientes comandos: dentro de nuestra carpeta que tiene el instalador de la base de datos **mongodb**:

```
mongod --bind_ip 127.0.0.1 --logpath  
C:\logs\mongo.log --logappend --dbpath  
c:\mongo\data --directoryperdb --install
```

--bind_ip: Comando que permite configurar la dirección IP, la misma que servirá como la dirección de nuestro base de datos.

--logpath: Parámetro que nos permitirá direccionar la ruta para los archivos de logs.

--logappend: Parámetro que nos indica el lugar de sobre escritura de nuestro logpath.

--directoryperdb: Parámetro que nos indica que, cada base de datos que cree se almacenara en un directorio diferente.

Luego nos aseguramos que el servicio de MongoDB este iniciado en servicios de Windows en este caso.

2. Se utilizó el mismo servidor independiente de plataforma, de software libre Xampp, donde contiene un servidor web (Apache), y un intérprete para lenguaje de script.(PHP), incluyendo un driver para la conexión con la base de datos no relacional (Mongo DB), en la siguiente tabla se indica las versiones de los software que se utilizaron:

Software	Versión
XAMPP	1.7.7
APACHE	2.2.21
PHP	5.3.8
MONGODB	1.1.4

Tabla 9: Versiones de Software

En la tabla se muestra las versiones del software utilizados para la aplicación con Base de Datos No Relacional

3. Para la carga de datos en una base de datos MongoDB, se realizó un programa en java que nos permitió migrar la información que estaba almacenada en una base de datos relacional (MySQL), donde podemos ver en el anexo 6, los mismos datos que fueron modelados en el formato JSON que se describe a continuación:

```

{
  "_id" : ObjectId("523e47c1ed92f2ed0a9d85ff"),
  "id_blog" : 34,
  "titulo" : " genealog\u009da de los dispo- sitivos yensamblajes que ",
  "categoria" : "astronomia",
  "fecha_creacion" : "2011-3-25",
  "entradas" :
  [
    {
      "id_entrada" : 6927,
      "texto_entrada" : "nzar a diario y de la gran seguridad queofrecen",
      "comentarios" :
      [
        {
          "id_comentario" : 150479,
          "comentario" : " ocupaba un lugar relevante la exhibici\u009d",
          "fecha_comentario" : "1981-7-23"
        }
      ]
    }
  ],
  "imagenes" :
  [
    {
      "ruta" : "images/258.jpg"
    }
  ]
}

```

4. A continuación, se procedió a la creación de nuestro proyecto, basándonos en los paquetes antes descritos, donde podemos ver en el [anexo 2](#).

4.2 NORMA ISO/IEC 9126

La norma ISO/IEC 9126 se enfoca a la calidad del producto y consta de las siguientes partes:

- Modelo de Calidad
- Métricas externas
- Métricas internas
- Calidad en el uso de métricas

Para evaluar y especificar un producto de software tenemos que definir las características de calidad adecuadas, teniendo en cuenta el objetivo del uso del producto que desean conseguir.

Qué son las normas ISO?

Son normas de calidad creadas por la Organización Internacional para la Estandarización (ISO) formada por estándares y guías relacionadas con sistemas de gestión, que se pueden aplicar en cualquier institución sin importar su tipo y herramientas específicas como los métodos de auditoría (verificación que los sistemas cumplan con el estándar).

Beneficios

- Disminución de efectos e incidencias en la producción y prestación servicios.
- Aumento de la productividad.
- Compromiso con los requisitos del cliente.
- Fácil acceso a grandes clientes y administraciones públicas.
- Mayor y mejor acceso a los mercados internacionales.²¹

Proceso de Certificación

Cada miembro de la ISO certifica a cada uno de los Organismos de Certificación para una auditoria correcta con sus recomendaciones, efectuada la auditoria cada uno de los miembros aprueba el registro para que el Organismo Certificador presente el certificado ISO (los certificados son emitidos por el Organismo Certificador o de Registro).

La empresa solicitante debe presentar una solicitud al Organismo de Certificación, con los datos de la empresa (tamaño de la compañía, cantidad y la localización de sus instalaciones, productos, cuáles de estos se incorporan al registro,

²¹

http://www.qualityteamconsulting.com/index.php?option=com_content&view=article&id=48&Itemid=58&lang=es

las personas de contacto para la ISO en la empresa, como se documentan y respaldan los procedimientos de acuerdo a los estándares de la Norma).

El siguiente paso es una evaluación por parte de los auditores preliminar del organismo contratado, dando lugar a indicaciones para la realización de un trabajo correcto, cumplida la corrección de la instancia se realiza una auditoría completa, si la empresa no es aprobada existen mecanismos para apelar la decisión.

Costos

El desarrollo e implementación de un SGC (Sistema de Garantía de Calidad), su costo se ve superado por las ganancias en eficiencia, productividad, rentabilidad, satisfacción del cliente y aumento de la presencia en diferentes mercados.

Los costos de una certificación varían de acuerdo al tamaño de la organización, la complejidad de sus procesos y la dispersión geográfica de sus operaciones, entre otras variables, se agrega los gastos previos a la preparación.

Porque algunas organizaciones fallan

El fracaso de algunas empresas en la implementación de un SGS y las normas ISO es la falta de compromiso.

Cumplir con los requisitos en la documentación requeridos por la norma no es suficiente, poner en marcha el sistema y dejarlo actuar sin una supervisión correcta, sin una definición en los objetivos de calidad, planificación, errores en la identificación, definición de los procesos, una incorrecta evaluación de los costos y beneficios, todo esto implica involucrar a los directivos en una atención adecuada a sus mercados, un liderazgo participativo, impulsando una cultura de calidad en la

organización, el esfuerzo compartido de los involucrados hará posible que los beneficios de trabajar según los estándares de la ISO se hagan realidad.

A continuación estudiaremos la Norma ISO/IEC 9126 que será aplicada en la implementación web.

Norma ISO/IEC 9126

La Norma ISO/IEC 9126 Software Product Evaluation (Evaluación de los Productos de Software) permite conocer las características de calidad y lineamientos para su uso correcto del software. Este estándar es útil para los desarrolladores, compradores, personal que asegure la calidad, evaluadores independientes, responsables de especificar y evaluar la calidad del producto software, valida la complejidad de los requisitos, habiendo identificado los requisitos de calidad, objetivos del diseño y pruebas, criterio del aseguramiento de la calidad.

La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta), este estándar define un marco conceptual de la calidad que considera los siguientes factores: Calidad del Proceso, Calidad del Producto de Software (Calidad Interna y Calidad Externa) y Calidad en Uso.

Características de la calidad Interna y Externa de la Norma ISO/IEC 9126

Funcionalidad

El software debe estar en la capacidad de facilitar las funciones que satisfagan las necesidades explícitas e implícitas, y para cumplir con los requisitos funcionales.

- **Idoneidad**
Saber si el software desempeña con las tareas correctamente.
- **Exactitud**
Evalúa el resultado final que obtiene el software y si tiene consistencia a lo que se espera de él.
- **Interoperabilidad**
Revisar si el sistema puede interactuar con otro sistema independiente.
- **Seguridad**
Verifica si el sistema puede impedir el acceso a personal no autorizado.

Usabilidad

El software debe de ser entendido, aprendido, usado y atractivo para el usuario.

- **Aprendizaje**
Determina que tan fácil es para el usuario aprender a utilizar el sistema.
- **Comprensión**
Determina que tan fácil es para el usuario comprender el funcionamiento del sistema.
- **Operatividad**
Determina si el usuario puede utilizar el sistema sin mucho esfuerzo.
- **Atractividad**
Verifica que tan atractiva se ve la interfaz de la aplicación.

Eficiencia

El software debe desempeñarse correctamente dependiendo de la cantidad de los recursos que están siendo utilizados y bajo las condiciones planteadas.

- **Comportamiento en el tiempo**
Verifica la rapidez en que responde el sistema.
- **Comportamiento de recursos**
Determina si el sistema utiliza los recursos eficientemente.

Mantenibilidad

Debe de ser capaz de adaptarse a nuevas especificaciones y requisitos del software.

- **Estabilidad**
Verifica si el sistema mantiene su funcionalidad con cambios realizados.
- **Facilidad de análisis**
Verifica si la estructura de desarrollo es funcional al momento de diagnosticar fallas.
- **Facilidad de cambio**
Verifica si el sistema puede modificarse fácilmente.
- **Facilidad de pruebas**
Evalúa si el sistema puede ser probado fácilmente.

Portabilidad

El software debe estar en la capacidad para poder trasladarlo de un lugar a otro según las necesidades.

- **Capacidad de instalación**
Determina si el software se puede instalar fácilmente.
- **Capacidad de reemplazamiento**
Determina la facilidad que el software puede ser reemplazado con otro similar.
- **Adaptabilidad**
Determina si el software puede trasladarse a otros ambientes.
- **Co-Existencia**
Determina si el software puede funcionar con otros sistemas.

Confiabilidad

El software debe de mantener un nivel de funcionamiento cuando está siendo utilizado bajo condiciones determinadas.

Capacidad de Mantenimiento

El software debe estar en la capacidad de poder ser modificado, estas pueden ser: correcciones, mejoramiento del mismo, adaptación a cambios en el entorno y detalles de los requerimientos funcionales.

Fiabilidad

Capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.

- **Madurez**
Verifica las fallas del sistema y si son eliminadas durante el tiempo de pruebas o uso del sistema.
- **Recuperabilidad**
Verifica si el software puede reasumir el funcionamiento y restaurar datos perdidos después de un fallo ocasional.
- **Tolerancia a fallos**
Verifica si la aplicación es capaz de manejar errores.

4.3 PRUEBAS DE RENDIMIENTO

Las pruebas de rendimiento pueden servir para diferentes propósitos, demostrar que se cumpla los criterios de rendimiento, comparar sistemas, medir la carga de trabajo, para lograr estos propósitos es necesario utilizar herramientas para determinar que tan eficaz es un sistema en el momento de realizar una tarea, validando y verificando los atributos de la calidad del sistema como la escalabilidad, fiabilidad y el uso de sus recursos.

Para alcanzar un nivel de rendimiento satisfactorio de un sistema nuevo, estas pruebas deben realizarse al comienzo del proyecto de desarrollo y durante la construcción del mismo.

Pruebas de carga

Esta prueba de rendimiento se realiza generalmente para ver el comportamiento de una aplicación dependiendo la cantidad de peticiones. Esta carga puede ser el número de usuarios utilizando la aplicación con el número de las transacciones que realicen durante el tiempo que dure la carga, así podemos observar los tiempos de respuesta de las transacciones importantes de la aplicación.

Prueba de estrés

Esta prueba determina la seguridad de la aplicación en los momentos de carga extrema, ayudando a los administradores a determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Prueba de estabilidad

Esta prueba determina si la aplicación puede aguantar una carga esperada continuada, es decir determina si hay alguna fuga de memoria en la aplicación.

Pruebas de picos

La prueba de picos, observa el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga, es recomendable realizarla con un software automatizado que permita realizar cambios en el número de usuarios mientras que los administradores llevan un registro.

4.4 COMPARACIÓN DE BASE DE DATOS NO RELACIONALES VS BASES DE DATOS RELACIONALES

En el siguiente cuadro se presenta un resumen comparativo entre las Bases de Datos Relacionales y no Relacionales, basándonos en el estudio realizado en el capítulo 1 y capítulo 3:

Descripción	No Relacional	Relacionales
Estructura de Datos	Es una estructura flexible, no es necesario definir una estructura de datos	Asumen una estructura bien definida de datos que son uniformes y las propiedades de estos datos pueden definirse por adelantado.
Relaciones entre tablas	No existen relaciones entre colecciones, no obstante puede depender del modelado de datos.	Tienen que estar muy bien establecidas y ser referenciadas de forma sistemática.
Transaccionalidad	Se pierde integridad en las transacciones.	Utilizan ACID
Consultas e Índices	Disminuye el uso de indexación y el poder de las consultas.	

Tabla 10: Comparación Base de Datos Relacional vs No Relacionales

4.5 RESULTADOS

A continuación se describe los materiales y las condiciones con las que se hicieron las pruebas de rendimiento, las mismas que detallaremos en los siguientes puntos.

4.5.1 MATERIALES Y CONDICIONES CON QUE SE REALIZARON LAS PRUEBAS DE RENDIMIENTO

Para realizar las pruebas de rendimientos para las dos aplicaciones realizadas se utilizó una portátil con las siguientes características, las mismas que se muestran en la tabla 8:

En Se utilizó una portátil como un servidor con las siguientes características, como se muestra en la siguiente tabla:

Características	Descripción
Procesador	Intel Core i3 de segunda generación.
Memoria Ram	6 GB
Tipo de Sistema	Windows 7 de 64 bits

Tabla 11: Características del Servidor

Se utilizaron las siguientes computadoras como clientes, con las siguientes características, como se muestra en las siguientes tablas:

Características	Descripción
Procesador	AMD Athlon(tm) 64 X2 Dual-Core
Memoria Ram	3 GB
Tipo de sistema	Sistema Operativo de 32 bits

Tabla 12: Características Cliente 1

Características	Descripción
Procesador	Intel(R) Celeron(R) de primera generación
Memoria Ram	2 GB
Tipo de sistema	Sistema Operativo de 64 bits

Tabla 13: Características Cliente 2

Como segunda condición se utilizaron diferentes tamaños de bases de datos, como se indica en la siguiente tabla:

Bases de Datos	Tamaño de la Base de Datos	Registros Blog
Base de Datos1	Aproximadamente 25 MB	50.000 registros
Base de Datos 2	Aproximadamente 50 MB	125.000 registros

Tabla 14: Tamaño de las Bases de Datos

4.5.2. Prueba de rendimiento aplicado a la visualización de la información en la aplicación web.

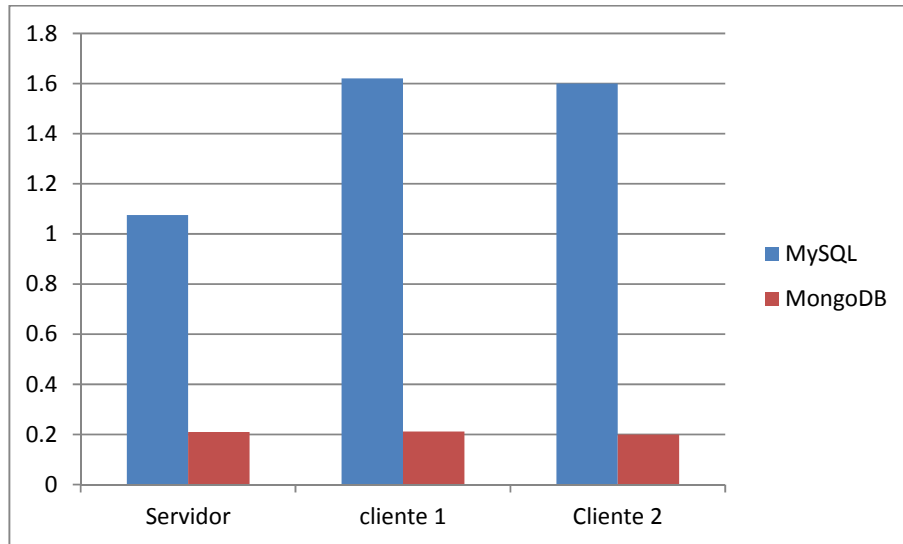
A continuación se muestran los resultados en gráficas de barra, obtenidas de los resultados de las consultadas realizadas en cada uno de las aplicaciones web que se muestran en la tabla 15 y tabla 16:

Considerando los datos que se muestra en la tabla 14, ejecutamos una primera prueba que cruza toda la información en una base de datos relacional y una base de datos no relacional generado por 49.545 registros; en el servidor, cliente 1 y cliente 2, los resultados se muestran en la siguiente tabla:

	Bases de Datos	Tiempo de Respuesta
Servidor	MySQL	1.07492495 milisegundos
	MongoDB	0.20933700 milisegundos
Cliente 1	MySQL	1.62012601 milisegundos
	MongoDB	0.21111703 milisegundos
Cliente 2	MySQL	1.6000234 milisegundos
	MongoDB	0.19930656 milisegundos

Tabla 15: Resultados con 49.545 registros

La gráfica que se muestra a continuación es el resultado de los 50.000 registros que se realizaron en las aplicaciones web tanto de MySQL como de MongoDB:



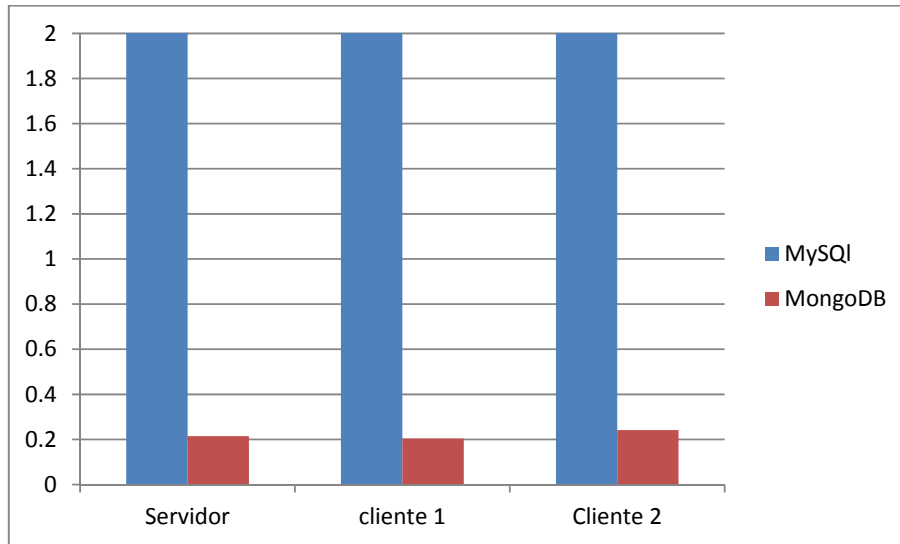
Gráfica 1: MySQL y MongoDB con 50.000 registros

Como segunda prueba cruzamos toda la información en una base de datos relacional y una base de datos no relacional generado por 125.264 registros; en el servidor, cliente 1 y cliente 2, los resultados se muestran en la siguiente tabla:

	Bases de Datos	Tiempo de Respuesta
Servidor	MySQL	146.62013197 milisegundos
	MongoDB	0.21517420 milisegundos
Cliente 1	MySQL	146.68430591 milisegundos
	MongoDB	0.20411611 milisegundos
Cliente 2	MySQL	146.91859889 milisegundos
	MongoDB	0.24168587 milisegundos

Tabla 16: Resultados con 125.254 registros

La gráfica que se muestra a continuación es el resultado de los 125.000 registros que se realizaron en las aplicaciones web tanto de MySQL como de MongoDB:



Gráfica 2: MySQL y MongoDB con 125.000 registros

En base a las pruebas realizadas, se establece que MongoDB tiene un mejor rendimiento en el tiempo de respuesta y visualización de los datos que MySQL, como se puede observar en las gráfica 1 y en la gráfica 2.

CAPÍTULO 5

CONDICIONES PARA EL USO DE LAS BASES DE DATOS NO RELACIONALES.

5.1 NORMATIVAS

Una base de datos no relacional va a depender de los requerimientos que la entidad requiera.

Una base de datos no relacional está ligada al manejo de grandes volúmenes de datos conocidos ya desde tiempo atrás como BIG DATA (grandes datos), además tienen que tener un alto porcentaje de procesamiento (rendimiento) y a la inclusión de diferentes formatos de datos (no homogéneos).

Una Base NoSQL no es un producto a pequeña escala sino una clase de productos con una colección de diversos conceptos en el almacenamiento y manipulación de datos.

Una Base de Datos NoSQL muy similar a una Relacional denominada es MongoDB ya que tiene una gran similitud en sus conceptos de modelado de datos.

La escalabilidad para que un sistema puede acoplar recursos adicionales para el aumento en su rendimiento.

La Base de Datos NoSQL puede trabajar con clústers siendo un conjunto de máquinas ordinarias que se utilizan comúnmente y que trabajaran como una sola unidad para trabajar de manera coordinada al momento que aumenta la carga de información.

Para escoger que tipo de Base de Datos es la adecuada para una aplicación, podemos tener como guía el teorema de CAP (o teorema de Brewer), nos dice que es imposible para un sistema de cómputo distribuido garantizar simultáneamente la consistencia, disponibilidad y tolerancia a fallos.²²

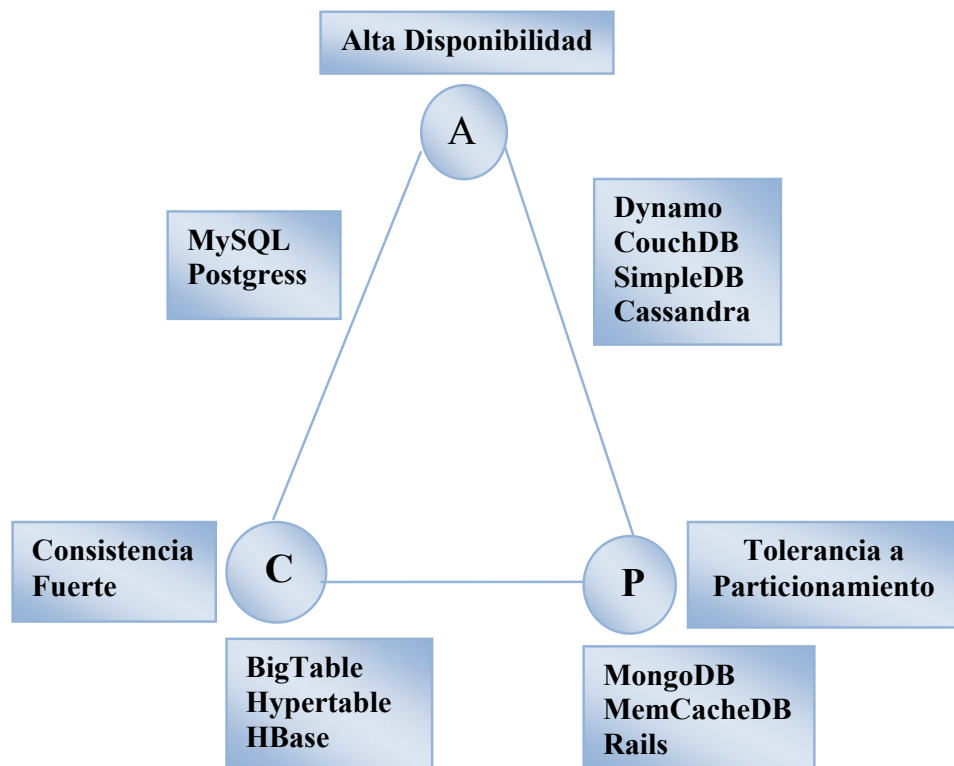


Figura 9: Teorema de CAP

²² http://es.wikipedia.org/wiki/Teorema_CAP

Según la figura 9, nos muestra la manera de cómo elegir una base de datos para realizar una aplicación dependiendo de los requerimientos, como se describe a continuación:

Para una aplicación consistente y tolerante a fallos se puede elegir entre las siguientes:

- BigTable
- Hypertable
- HBase
- MongoDB
- MemCacheDB
- Rails

Para una aplicación que requiera de consistencia y disponibilidad de datos se puede elegir las siguientes:

- MySQL
- Postgres

Para una aplicación que requiera consistencia y tolerancia a fallos se puede elegir entre las siguientes:

- Dynamo
- CouchDB
- SimpleDB
- Cassandra

CONCLUSIONES

Al momento de realizar las dos aplicaciones nos pudimos dar cuenta de muchos factores que intervinieron para su desarrollo entre los cuales mencionamos a continuación:

Se determinó que al usar una base de datos no relacional en una aplicación web no transaccional, al momento de realizar las consultas su tiempo de respuesta en la visualización de los resultados disminuye considerablemente, por la gran cantidad de información que tiene la base.

A través de una investigación sobre las bases de datos no relacionales, se identificó algunas funcionalidades importantes, entre estas tenemos: son capaces de procesar grandes cantidades de información, se puede diseñar diferentes aplicaciones dependiendo de los requerimientos del usuario, son aplicables a la tecnología existente (lenguajes de programación).

El principal beneficio que se obtiene al momento de utilizar una base de datos no relacional es la flexibilidad de almacenar los datos, si se requiere de un campo extra no es necesario cambiar la estructura de nuestra base de datos como lo que ocurriría en una relacional.

Se pudo comprobar que el tamaño de las bases de datos no relacionales es mucho más grande que una base de datos relacional y aun así en los tiempos de ejecución son más eficiente, teniendo un margen de tolerancia a fallos como: redundancia de datos.

RECOMENDACIONES

En esta sección citaremos algunos aspectos a tener en cuenta en el desarrollo de proyectos con bases de datos relacionales y no relacionales:

Al realizar este tipo de proyectos debemos tener en cuenta que se debe de realizar un análisis bastante minucioso, es vital determinar cada uno de los requerimientos y sus características para de esta manera se pueda cubrir con las necesidades requeridas para de esta manera evitar gastos innecesarios.

Se debe desarrollar un buen diseño de la Base de Datos para que la información no contenga datos redundantes, en el caso de una base de datos relacional.

Es importante saber que al momento de utilizar software libre para disminuir costos y mantener un buen rendimiento, se debe tener en cuenta la capacidad que deben tener estos programas para el buen manejo de la información.

BIBLIOGRAFÍA

- SILBERSCHATZ, Abraham, Fundamentos de Bases de Datos, Cuarta Edición, 2002.
- SHASHANK, Tiwari, Professional NoSql, 2011.
- ROEBUCK, Kevin, Storing and managing big data NoSQL, Hadoop and more.
- Oracle Database 11g Release 2: Manageability and Real Application Testing Overview.
- Lic. Elda Monterros, Normas ISO,
<http://www.unlu.edu.ar/~ope20156/normasiso.htm>
- 05-septiembre-2013 http://es.wikipedia.org/wiki/Modelo_relacional.
- Ken Hess, Top 10 Enterprise Database System to Consider,
<http://www.serverwatch.com/print.php>
- 31-agosto-2-13, Sistemas Operativos GN, <http://www.gnu.org/philosophy/free-sw.es.html>
- Php Language Center <http://docs.mongodb.org/ecosystem/drivers/php>
- Source Forge, <http://sourceforge.net/projects/xampp/files>
- Netbeans, <https://netbeans.org/downloads/>
- MongoVue, <http://www.mongovue.com/>
- WebYog, <https://www.webyog.com/product/downloads>
- MongoDb <http://www.mongodb.org/downloads>
- Mysql, <http://www.mysql.com/>

ANEXOS

Anexo 1



Anexo 2

The screenshot shows a Mozilla Firefox browser window with two tabs: 'Blog MySQL' and 'Blog MongoDB'. The address bar shows 'localhost/tesis/index.php'. The page content includes the header 'Universidad Politécnica Salesiana' and the 'mongoDB' logo. A search bar on the left contains the text 'monologicas' and a 'Buscar' button. Below the search bar, it displays 'Tiempo: 0.21800613 milisegundos'. The search results area shows a snippet: 'ciones monologicas tanto de la economia politica,' followed by 'acuatico (2001-5-26)' and 'Posted by Someone'. A small image of a red vintage car is visible below the text. The Windows taskbar at the bottom shows various application icons and the system tray with the date '30/09/2013' and time '09:32'.