

UNIVERSIDAD DE HUANUCO

FACULTAD DE INGENIERÍA

E.A.P. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



TRABAJO DE SUFICIENCIA PROFESIONAL

**“ANÁLISIS COMPARATIVO ENTRE BASE DE DATOS
RELACIONALES Y BASE DE DATOS NOSQL”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE
SISTEMAS E INFORMÁTICA**

Autor:

Bach. Pedro Luis Gonzales Linares

ASESOR

ING. LUIS ANDRES, MEZA ORDOÑEZ

**HUÁNUCO – PERÚ
2016**



UNIVERSIDAD DE HUANUCO

Facultad de Ingeniería

E.A.P. DE INGENIERÍA DE SISTEMAS E INFORMATICA

ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO(A) DE SISTEMAS E INFORMÁTICA

En la ciudad de Huánuco, siendo las...18:30... horas del día...06... del mes de...diciembre... del año...2018..., en el Auditorio de la Facultad de Ingeniería, en cumplimiento de lo señalado en el Reglamento de Grados y Títulos de la Universidad de Huánuco, se reunieron los Jurados Calificadores Nombrados mediante la Resolución N° ...1153-2018-D-FI-UOH... integrado por los docentes:

ING. PAOLO EDUER SOLIS JARA..... (Presidente)

ING. JOSE ANTONIO NÚÑEZ VICENTE (Secretario)

ING. ETHEL JHOVANI MANZANO LOZANO (Vocal)

Para calificar el Trabajo de Suficiencia Profesional solicitado por el (la) Bachiller Pedro Luis Nicolás, Gonzales Linarez para optar el Título Profesional de Ingeniero(a) de Sistemas e Informática.

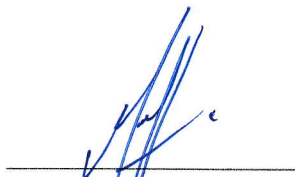
Dicho acto de sustentación se desarrolló en dos etapas: exposición y absolución de preguntas: precediéndose luego a la evaluación por parte de los miembros del Jurado.

Habiendo absuelto las objeciones que le fueron formuladas por los miembros del Jurado y de conformidad con las respectivas disposiciones reglamentarias, procedieron a deliberar y calificar, declarándolo (a)..Aprobado por..Unanimidad.. con el calificativo cuantitativo de...14... y cualitativo de..Suficiente..

Siendo las...19:40... horas del día...06... del mes de...diciembre... del año...2018..., los miembros del Jurado Calificador firman la presente Acta en señal de conformidad.



Presidente



Vocal



Secretario

DEDICATORIA

Primeramente, agradezco a Dios quien me guía, me dio la fortaleza y perseverancia necesaria para alcanzar mis metas y llegar hasta acá, a mis Padres, que son fundamentales, porque me inculcaron principios, valores, perseverancia, coraje, responsabilidades y en especial mucho amor, y por sus enseñanzas que me dieron.

AGRADECIMIENTO

Un agradecimiento especial a mi madre, que nunca dejo de apoyarme, hasta en las cosas más mínimas siempre estuvo a mi lado ayudándome, preocupada y dándome mucha motivación para seguir y así poder concluir mi carrera con éxito, gracias a ella es que seguí adelante.

A mi padre, que con su comprensión me dio ánimos para no rendirme, es mi inspiración.

A mis docentes de la Facultad, que me ofrecieron sus conocimientos para mi formación académica y con ello me formaron como un buen ingeniero de sistemas.

RESUMEN

Existe un prejuicio que indica que las bases de datos NOSQL, no son competentes frente a las SQL, asociado al poco interés o poca información de los profesionales de TI sobre las bondades de las bases de datos orientadas al modelo NOSQL, y, porque no, el creer que las bases de datos NOSQL, se encuentran en una fase de inicio, esto genera la duda que produce esta investigación.

El proyecto de investigación busca hacer una comparación, entre las bases de datos SQL y las bases de datos NOSQL.

Esta investigación empieza señalando un preámbulo de las bases de datos SQL, además de hacer un diagnóstico individual del modelo de base de datos relacional y no relacional, y así reflejar una serie de comparaciones entre ambos modelos que involucre varios criterios de evaluación entre dichos modelos.

La contribución consiste en un análisis comparativo de modelos de datos, que nos permite tener una referencia para el cambio de terminología de base de datos relacional hacia la no relacional mediante la determinación de diferencias y similitudes. Ello es a partir de propiedades y patrones en los modelos de datos: restricciones, estructura y operaciones.

Los indicadores para dicho análisis de tiempos de respuesta, a través de escenarios en unos casos, revelan que una Base de datos No SQL, en muchos de los contextos, es más óptima por sus características adjuntas y el desempeño que tiene sobre un determinado ambiente. Por otro lado, de la escalabilidad, evidenciada a través de las variaciones de carga de data y hardware, resulta ser superior el producto No SQL ante su contendiente relacional.

ABSTRACT

There is a prejudice that indicates that the databases oriented to the non-relational model are not competent versus non-relational ones, associated to the little interest or little information of the IT professionals regarding the benefits of the NOSQL databases, and, because no, considering that the databases oriented to the non-relational model are only in an initial phase, have generated the dilemma that this research produces.

The research project seeks to carry out a comparative analysis between relational databases and non-relational databases, which will help to broaden a point of view related to the potentialities of non-SQL databases.

The development of this research begins by pointing to an introduction to the No SQL databases, in addition to making an individual diagnosis of the relational and non-relational data model, and thus reflect a comparative analysis between both models that involves a series of evaluation criteria between said models.

The contribution is a comparative analysis of data models, which allows us to have a useful reference for the change from relational to non-relational terminology through the determination of equivalences and differences. This is based on properties and standards in the data models, restrictions, structure and operations.

The indicators for analysis of response time, through scenarios in some cases, reveal that a non-SQL DB, in many of the contexts, is more optimal due to its attached characteristics and the performance it has over a given environment. On the other hand, of the scalability, evidenced through the variations of data and hardware load, it turns out to be superior the No SQL product before its relational contender.

INDICE

DEDICATORIA.....	II
AGRADECIMIENTO.....	III
RESUMEN.....	IV
ABSTRACT.....	V
INTRODUCCIÓN.....	XI
CAPÍTULO I.....	12
1.-Problema de investigación.....	12
1.1. Descripción del problema.....	12
1.2. Formulación del problema.....	14
1.2.1. Problema general.....	14
1.2.2. Problemas específicos.....	14
1.3. Objetivos.....	14
1.3.1 Objetivo General.....	14
1.3.2 Objetivos específicos.....	14
1.4. Justificación de la Investigación.....	14
1.5. Limitaciones de la Investigación.....	15
1.6. Viabilidad de la Investigación.....	15

1.7. Variables	16
1.7.1. Variable Independiente.....	16
1.7.2. Variable Dependiente	16
1.7.3. Operacionalización de Variables (dimensiones e indicadores) ..	16
 CAPÍTULO II.....	 18
2.-Marco Teórico	18
2.1. Antecedentes de la Investigación	18
2.2. Bases Teóricas.....	24
2.2.1. Base de Datos.	24
2.2.2. Gestor de Base de Datos.....	25
2.2.5.-Tipos de Bases de Datos.....	27
2.2.6.-Historia de las Bases de Datos.....	28
2.2.7. SQL.....	31
2.2.8. Gestores de Bases de Datos Relacionales.....	32
2.2.9. NO SQL.....	33
2.2.10. Bases de Datos SQL vs Bases de Datos NOSQL.....	38

2.2.11. Diferencias entre SQL y NOSQL.....	39
2.2.12. Criterios de uso de Base de Datos SQL.	39
2.2.13. Criterios de uso de Base de Datos NOSQL.	40
2.2.14. Ejemplos de bases de datos NOSQL.	41
2.2.15. Compañías que usan NOSQL.	43
2.2.16. Big data.....	43
2.3. Definiciones conceptuales.....	44
CAPÍTULO III	47
3. Metodología de la Investigación	47
3.1. Tipo de investigación (Referencial).....	47
3.1.1. Alcance o nivel	47
3.1.2. Diseño.....	48
3.2. Técnicas e instrumentos de recolección de datos.....	48
3.3. Técnicas para el procesamiento y análisis de la información...48	
CAPÍTULO IV	49
4.1. Procesamiento de datos.....	49

4.2. Criterios de uso de base de datos SQL Y base de datos NOSQL	51
CONCLUSIONES.....	54
RECOMENDACIONES	55
5. BIBLIOGRAFÍA.....	56

INDICE DE TABLAS

Tabla 1.

Diferencias entre las bases de datos relacionales y no relacionales49

Tabla 2.

Comparaciones de bases de datos relacionales y no relacionales50

Tabla 3.

Características de ACID y BASE51

Tabla 4.

Diferencias ente SQL y NOSQL.....53

INTRODUCCIÓN

En la actualidad, las bases de datos SQL o relacionales se proyectan a ser una de las herramientas de mayor uso en nuestra sociedad, las cuales permiten manipular, grabar y obtener información de distintos campos. Sin embargo, han debido pasar varios años para que, las bases de datos SQL tengan el desarrollo que hoy tienen, comenzando con el almacenamiento de data en tarjetas perforadas, cintas magnéticas y uso de discos que han dado comienzo a las bases de datos que hoy conocemos. Surge entonces el modelo relacional, que principalmente consiste en series de reglas para evaluar a los administradores de sistemas de datos relacionales

El modelo tuvo baja acogida principalmente por tener un poco rendimiento a comparación de otros sistemas de almacenamiento. Pues estos sistemas utilizaban tablas para almacenar data, excluyendo de elementos primordiales como son de claves primarias, etc.

CAPÍTULO I

1.-Problema de investigación

1.1.Descripción del problema

Actualmente las bases de datos más conocidas y utilizadas por muchas instituciones y empresas son las del tipo relacional, que son una buena alternativa en muchos casos, sin embargo, al manejar una mucha data lo más conveniente sería utilizar una base de datos NOSQL.

El almacenado de datos se había venido desarrollando de manera habitual por las bases de datos tradicionales desde 1970 Edgar Fram Codd propuso el primer modelo relacional, aunque es Oracle quien le da el posicionarlo de manera comercial más adelante esas BD son empleadas por el lenguaje de consultas estructuradas DQL que permite, manejar la información que a su vez es almacenada en forma estructurada. El desarrollo de este tema es basado en acumular bibliografía y documentos, aplicando y presentando un estudio de comparaciones entre el rendimiento de las bases de datos no relacionales frente a las bases de datos relacionales (tradicionales) demostrando un estándar que nos muestre la facilidad de manejar datos en NOSQL a través de análisis que nos ayude a obtener datos e interpretarlos.

La problemática actual es que muchas empresas implementan soluciones sin criterio, generando pérdidas de dinero y tiempo en sus organizaciones y a la vez existe una nueva tecnología de BD, que permite manejar un mayor número de información.

Por lo expuesto, esta tesis busca darnos los criterios necesarios para elegir entre las relacionales y las NOSQL. Conociendo los beneficios y limitaciones de las bases NOSQL, los desarrolladores de base de datos

evalúan las situaciones donde se pueden aprovechar los beneficios que estas tienen sobre las bases de datos SQL tradicionales en determinados ámbitos.

1.2. Formulación del problema

1.2.1. Problema general.

¿Cuáles son los criterios a considerarse para un análisis comparativo entre una BD Relacional y una BD NOSQL?

1.2.2. Problemas específicos.

P.E.1 ¿en qué casos se recomienda usar bases de datos relacionales?

P.E.2. ¿en qué casos se recomienda usar bases de datos NOSQL?

1.3. Objetivos

1.3.1 Objetivo General.

Determinar cuáles son los criterios a considerarse para realizar análisis y comparar una Base de Datos Relacional y una Nosql

1.3.2 Objetivos específicos.

O.E.1.Determinar los casos en los que se recomienda usar bases de datos relacionales.

O.E.2.Determinar los casos en los que se recomienda usar bases de datos NOSQL.

1.4. Justificación de la Investigación

La justificación en este proyecto es de tipo teórica ya que se dará un panorama amplio para la elección de criterios de uso de Bases de Datos Relacionales y Nosql.

Las bases de datos no relacionales, son populares en grandes empresas como Google , Facebook, Twitter que vieron la necesidad de

guardar grandes cantidades de información, obtuvieron aciertos, como aplicabilidad, de rendimiento alto ,hoy en día, hay un nuevo estudio que viene ganando reconocimiento por su versatilidad y escalabilidad, son las bases de datos no relacionales , ya que no obligan a hacer estructuras de datos en forma de relaciones y tablas sino que manejan columnas por ello son más flexibles y se puede guardar la información con ventaja de escalabilidad ,que refiere al proceso de análisis de múltiples datos. Se puede ver que esta tecnología sirve de gran manera para manejar los datos y su aplicación se expande a distintos sectores de la comunidad, donde el manejo de información es primordial.

Este proyecto consta de una recopilación de criterios para comparar entre una base de datos Relacional y una Nosql, tiene una relevancia teórica para futuras investigaciones, ya que este tema es poco tratado en nuestro entorno, Así también, sirve brinda una utilidad práctica, proporcionando criterios para que los usuarios puedan escoger entre usar una base de datos relacional y/o una base de datos NOSQL.

1.5. Limitaciones de la Investigación

Este proyecto tiene como limitación la poca información acerca del tema en las páginas y libros, por ser este un tema nuevo.

1.6. Viabilidad de la Investigación

Será posible realizar la presente investigación con el refuerzo de los recursos con los que se cuentan, así como el apoyo técnico de otros profesionales que saben acerca del tema, en conclusión, el proyecto es viable, por el lado del costo del proyecto, este es mínimo por el fácil acceso a la información con los aportes del internet y también de los textos de las bibliotecas y con la cantidad de información al alcance de nosotros será posible realizar el presente proyecto.

1.7. Variables

1.7.1. Variable Independiente

Criterios de uso

1.7.2. Variable Dependiente

Análisis comparativo exitoso de la base de datos.

1.7.3. Operacionalización de Variables (dimensiones e indicadores)

OBJETIVO GENERAL: Determinar cuáles son los criterios a considerarse para el análisis comparativo entre una Base de Datos Relacional y una NOSQL.

Variables	DEFINICION	DIMENSION	Indicadores
CRITERIOS A CONSIDERAR PARA EL ANALISIS DE UNA BASE DE DATOS RELACIONAL	PRINCIPALES CARACTERISTICAS CON LAS QUE PODEMOS DIFERENCIAR A LAS BASES DE DATOS RELACIONALES	CARACTERISTICAS	Durabilidad
		VENTAJAS	Consistencia
		DESVENTAJAS	Atomicidad
			Integridad de datos
CRITERIOS A CONSIDERAR PARA EL ANALISIS DE UNA BASE DE DATOS RELACIONAL	PRINCIPALES CARACTERISTICAS CON LAS QUE PODEMOS DIFERENCIAR A LAS BASES DE DATOS NOSQL	CARACTERISTICAS	La escalabilidad
		VENTAJAS	Flexibilidad
		DESVENTAJAS	Velocidad

Variables	Indicadores
Implementación exitosa de una BD	Redundancia mínima
	Multi acceso
	Seguridad
	Integridad de datos
	Respaldo
	Velocidad consultas

CAPÍTULO II

2.-Marco Teórico

2.1. Antecedentes de la Investigación

2.1.1.-Tesis Internacional 1.

Bas (2015), “Estudio comparativo de BBDD relacionales y NOSQL en un entorno industrial”, en la Escuela Técnica Superior de Ingeniería Informática Universidad Politécnica de Valencia.

Haciendo uso de bases de datos NOSQL, buscaremos la forma de un incremento en el rendimiento de la aplicación en el momento de mostrar datos al usuario sin la escasez de modificar el hardware del que se dispone actualmente.

Conclusiones:

Durante este trabajo hemos comparado los distintos puntos a valorar en un sistema de bases de datos. El objetivo de éste, era encontrar un sistema gestor con el que mejorar el rendimiento del actual sistema SQL. En el capítulo 4 comparamos los mejores gestores NOSQL del mercado para buscar el que se adaptase a nuestros requisitos de forma más conveniente. En posteriores capítulos se ha analizado el formato de datos en el sistema NOSQL y el acceso a estos. En estos capítulos, hemos buscado en todo momento la forma más rápida de guardar y acceder a los datos. Con este objetivo se trazó un plan para comparar diferentes sistemas gestores en la mayor igualdad de condiciones. Como resultado de este plan hemos podido deducir que las bases de datos NOSQL, con Couchbase como representante en este proyecto son idóneas para este tipo de almacenes. Los resultados sacan a relucir las carencias de los sistemas gestores SQL a la hora de gestionar múltiples tablas,

mientras que ensalzan las características “bigdata” de Couchbase. Los sistemas NO SQL se han mostrado mucho más eficientes en el tiempo de gestionar grandes cantidades de datos. Mientras más grande era el número de variables y el periodo de selección de datos, mejor rendimiento ofrecía respecto a sus homólogos SQL. Existen varios aspectos que por alcance no se han podido contemplar en este trabajo y que podrían ofrecer una visión más amplia y real a la hora de comparar los diferentes sistemas. Como sabemos, en un entorno de producción actual para un proyecto de relativa envergadura nunca habrá un único servidor de bases de datos. Por ello, en proyectos como podría haber sido Regedate, pueden existir varios servidores, o incluso se pueden encender y apagar servidores en proporción de la demanda al sistema. De este modo, las peticiones no se gestionan en un único servidor como hemos visto en este proyecto, se gestionan sobre múltiples servidores. Al repartir el trabajo sobre estos diferentes servidores no solo se consigue agilizar los picos de trabajo. Se mejorará enormemente la obtención de datos. Y es aquí donde haremos especial hincapié en la funcionalidad map/reduce, ya que esta está expresamente preparada para este tipo de entornos y es capaz de multiplicar el rendimiento. Obviamente sobre SQL también existe la posibilidad de dividir el servidor en múltiples sub-servidores, pero estos de ninguna forma serán tan flexibles y rápidos como los servidores NOSQL. Por los motivos citados y por la alta tasa de transmisión de datos, en este proyecto se ha considerado a los sistemas NOSQL y en concreto a Couchbase como mejor posicionados para este tipo de aplicaciones donde se manejen un alto número de datos con pocas relaciones.

Los aspectos relacionados con el proyecto de investigación son que, haciendo uso de bases de datos no relacionales, buscaremos la base de incrementar el rendimiento de la aplicación en el momento de mostrar datos al usuario sin la necesidad de modificar el hardware del que se dispone actualmente.

2.1.2.-Tesis Internacional 2.

Mancilla (2013), Universidad de san Carlos de Guatemala, Uso de base de datos NOSQL documentales para crear sitios web de alto beneficios.

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

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

- ✓ La implementación de MongoDB 10gen supone ahorro en gastos de infraestructura para la empresa.

- ✓ Los aspectos relacionados con el proyecto de investigación son que, será posible el adelanto de sitios web de alto rendimiento utilizando base de datos NOSQL documentales

2.1.3.-Tesis Nacional 1.

Mendoza, Medina (2016),” NO SQL FUTURO ALMACENAMIENTO DE DATOS”, Universidad Peruana de Ciencias Aplicadas – UPC.

El prototipo de herramienta es la materialización o implementación en una aplicación la propuesta de procedimiento de migración. Está sigue, en sí, las reglas o equivalencias que entre modelos de datos que se definieron como propuesta. Indicadores comparativos de tiempo de respuesta y escalabilidad El análisis comparativo de tiempo de respuesta y escalabilidad se estructuró a partir de someter a determinadas bases de datos de los modelos estudiados a distintos escenarios ensayados de tiempo de respuesta y escalabilidad sobre data existente. A partir de estos, se obtuvieron medidas o cifras que pudieron revelar tendencias o estadísticas entre los productos pertenecientes a cada modelo. Las dimensiones empleadas en las gráficas resultantes fueron las siguientes:

- ✓ Cantidad de queries versus Tiempo (segundos)
- ✓ Cantidad de comandos versus Tiempo (segundos)
- ✓ # De operaciones Insert versus Tiempo (segundos)
- ✓ # De operaciones Delete/Remove versus Tiempo (segundos)

✓ El diagnóstico individual de cada modelo de datos a nivel de las propiedades estándares en estos, tales como estructura, restricciones y operaciones, brindan un punto de partida por el cual se pueden establecer equivalencias y diferencias, y así lograr conseguir sinergias o una comunicación a pesar de existir múltiples modelos de datos.

✓ Realizar un análisis comparativo entre ambos modelos de datos, permitió evidenciar las diferencias que muestran estos, y, además, encontrar equivalencias que permitieron una comunicación de un modelo hacia otro. En esta investigación se empleó un modelo de datos documental para la parte de las no relacionales, y se puede asegurar que a nivel de restricciones existen grandes diferencias con su contraparte relacional, ya que a consecuencia de la estructura no rígida que presenta impide pueda respetarse limpiamente tal propiedad.

✓ Materializar una herramienta prototipo basada en una propuesta de procedimiento de migración que identifica, sobre todo para este propósito, equivalencias entre dos distintos modelos de datos, es una ventaja que permite, además de poner en iguales condiciones a dos bases de datos de diferentes modelos para los propósitos de esta investigación, lograr integrar tecnologías relativamente distintas.

✓ Haber comprobado el buen tiempo de respuesta y escalabilidad del modelo de datos no relacional en un producto NoSQL, a través de escenarios en una prueba de concepto, de alguna manera, contribuirá a incrementar el interés en la comunidad de profesionales TI.

✓ Haber obtenido, por ejemplo, 6.94 seg de diferencia en un escenario a favor del modelo no relacional sobre el volumen que bordeó el millón de datos, permite deducir que, si habláramos de tiempos de respuesta masivos análogas a los que realizan conocidas aplicaciones a gran escala, estaríamos

discutiendo de 6.94 segundos multiplicados por la capacidad de datos involucrados; es decir, en promedio horas que serían aprovechadas. NoSQL: futuro de almacenamiento de datos.

✓ La situación actual de las NoSQL, es muy importante para entender las razones principales por las que estas, hoy en día, están revolucionando en situaciones particulares y nuevas necesidades de los negocios. Y, sin duda, esta nueva tendencia está aperturando el mercado de innovación para las empresas, en cuanto, precisamente, a nivel de base de datos, asegurar un esquema flexible, tiempo de respuesta y escalabilidad.

✓ El estudiar sobre trabajos similares en el campo de las tecnologías de base de datos no relacionales, se basó, también, en demostrar nuestra creencia relacionada a que ningún modelo de datos, ya sea el relacional o no relacional, es superior uno respecto a otro, sino querer probar que las NoSQL, a través del proceso detallado mostrado a lo largo de este trabajo y, naturalmente, de los estudios mencionados que comparten el ideal, son ya, altamente, capaces de ser utilizadas en ambientes de producción y poder sobrevivir como complementos mutuos, según sea el caso, con las del modelo relacional.

En el periodo de recopilación de antecedentes, solo he podido encontrar 1 tesis nacional referente a mi trabajo por esta razón no pude poner las 3 tesis nacionales como indica la norma APA.

Aspectos relacionados al tema: sobre este antecedente de la investigación, se puede ver que se hace un prototipo para la migración de una base de datos Relacional a una base de datos Nosql, para así ver cuál es la capacidad, velocidad de respuesta y otras ventajas de este modelo de base de datos.

2.2. Bases Teóricas

En este ítem daremos un panorama amplio de todo el proyecto con conceptos que nos ayuden a realizarlo.

2.2.1. Base de Datos.

Base de Datos es un vínculo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles a tiempo real y compatibles con usuarios asistentes con necesidad de información diferente y no predicable en tiempo. (Valdés, 2007)

✓ Tabla

Una tabla es un objeto que puede almacenar data en filas y columnas. Las filas se denominan registros y las columnas campos. La data almacenada en una tabla se refiere a un tema determinado dentro de la base de datos, por ejemplo, datos de los libros, autor, índice, y diferentes años y materias principales. Dichas Bases de datos están determinadas por lo consistentes y lógicas que sean las tablas que implementemos.

✓ Consultas

Las consultas se utilizan para rescatar la información almacenada en las tablas. Con esta descripción podríamos pensar... ¿y por qué no la miramos directamente en ellas? Pues bien, la ventaja se encuentra en la posibilidad que ofrecen las consultas de seleccionar la información y exponer sólo aquellos datos que interesen en cada caso.

Las consultas se hacen a partir de expresiones que nos ayudaran a relacionarnos con la base de datos para extraer información de una o muchas tablas.

✓ Formularios

Un formulario es un formato usado para modificar, adicionar o consultar información bajo criterios del usuario.

Los formularios proveen listas desplegadas, gráficos instrucciones y controles de desplazamiento y que ayudan a los usuarios a trabajar con los datos. De un modo u otro, los formularios hacen que el trabajo sea más fácil.

✓ Informes

Un informe es usado para imprimir los registros guardados en una base de datos, utilizando un formato personalizado por el usuario. Los informes permiten mostrar totales para los grupos o informes completos, agrupan registros, etc.

Normalmente, los informes se construyen a través de los resultados obtenidos de la ejecución de consultas. De esta forma unimos la posibilidad de seleccionar sólo los datos que deseamos que nos ofrecen las consultas con la ventaja de imprimirlos. (ALCALDE, EL BAUL DEL PROGRAMADOR, 2016)

2.2.2. Gestor de Base de Datos.

Un gestor de base de datos o un SGBD (sistema de gestión de base de datos), nos permite, organizar, introducir y recuperar la información, el más utilizado es el modelo relacional el cual se encarga de guardar los datos en forma de tablas.

Las principales características de los gestores (aunque sean distintos) son:

- ✓ admitir definiciones de esquemas y vistas

- ✓ manipulan los datos y siguiendo las órdenes de los usuarios.
- ✓ cuidan que se respete la seguridad e integridad
- ✓ permite definir usuarios y restricciones de acceso
- ✓ controlan la concurrencia y las operaciones asociadas a la recuperación de los fallos.

2.2.3. Funciones de un SGBD.

Ejecutar las operaciones sobre la base de datos para proporcionarlos al usuario en función de su pedido se realiza de un modo eficiente y seguro. las características de un SGDB hacen posible el cumplimiento de funciones, que pueden agruparse de la siguiente manera:

1. Definición de los Datos. El SGBD define todos los objetos de la base de datos partiendo de definiciones en versión fuente para convertirlas en la versión objeto.

2. Manipulación de los Datos. El SGBD responde a las solicitudes del usuario para la realización de operaciones de borrado, actualización, extracción, entre otras gestiones. El manejo de los datos se realiza de una forma rápida, según las peticiones realizadas por los usuarios, y realizar la modificación del esquema de la base de datos.

3. Seguridad e Integridad de los Datos. aplica las medidas de seguridad e integridad de los datos adopta medidas que garantizan su utilidad, previamente definidas. Un SGBD debe respaldar su seguridad frente a ataques o impedir su acceso a usuarios que no tengan autorización por cualquier razón.

4. Recuperación y Restauración de los Datos. La restauración y recuperación de los datos ante un fallo es otra de las principales funciones de

un SGBD. Su aplicación se realizará a través de un Plan de restauración y recuperación de la información que sirva de respaldo.

2.2.4. Tipos de gestores de Bases de Datos.

La tipología de los SGBD varia, en función a los criterios que utilicemos para clasificarlos. Agruparlos atendiendo al modelo de datos, número de sitios o usuarios suele ser lo más habitual:

Si atendemos al modelo de datos, los gestores de bases de datos pueden ser:

- Relacionales
- En Red
- Jerárquicos
- Orientados a objetos

Es posible diferenciarlos según sean o no dueños, en función de la licencia, de acuerdo con el número de usuarios monousuario o multiusuario y, por ejemplo, también agruparlos en centralizados y distribuidos, esta vez según el número de sitios.

2.2.5.-Tipos de Bases de Datos.

En un principio existían tres tipos de bases de datos según su estructura interna (la manera de organizar la información):

- Bases de datos jerárquicos
- Bases de datos en red
- Bases de datos relacionales

- Bases de datos no relacionales

Más recientes son las bases de datos orientadas a objetos y las bases de datos multidimensionales.

A. Bases de Datos Jerárquicos.

Una base de datos jerárquica es un sistema de gestión de bases de datos que, recolecta la información en estructuras jerárquicas, que unen los registros en forma de estructura de árbol (como un árbol visto al revés), en donde un *nodo* padre de información puede tener varios nodos hijo. (INFORMACION, 2016)

B. Bases de Datos en Red.

Es una organización jerárquica de nodos, pero un nodo hijo puede tener más de un solo nodo padre (relación muchos a muchos). En estas bases de datos, existen los punteros, que son enlaces entre nodos padres y nodos hijos, que permiten acceder a un nodo por vías distintas accediendo al mismo en dirección descendente por las diversas ramas. (ALFONSO, S.F, pág. 1)

C-Base de Datos Relacional.

La base de datos relacional es un tipo de base de datos que cumple con el modelo relacional (el modelo más usado actualmente para implementar las bases de datos ya planificadas).

Permite establecer 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". (ALCALDE, EL BAUL DEL PROGRAMADOR, 2016)

2.2.6.-Historia de las Bases de Datos.

La primera vez que se escuchó este término, fue en un congreso celebrado en California en 1963. Una BD es un conjunto de información que se encuentra reunida o estructurada.

Sus orígenes están en la antigüedad, donde ya existían bibliotecas y toda clase de registros, también la utilizaban para acumular información sobre las cosechas.

Comenzaron a desarrollarse cuando empezó a ser necesario recopilar grandes cantidades de información o datos. El concepto de bases de datos está ligado a la informática desde hace mucho tiempo.

En 1884 Herman Hollerith creó la máquina automática de tarjetas perforadas, ante este hecho empezó a crear una máquina tabuladora, basada en tarjetas perforadas.

En la década de los 50 se dio origen a las cintas magnéticas, esto sirvió para sustituir las necesidades de información de las nuevas industrias.

En la década de los 60, bajaron los precios de los ordenadores para que se puedan adquirir y popularizarse el uso de los discos. En dicha época también comenzaron las primeras generaciones de bases de datos de red y las bases de datos jerárquicas. Durante este tiempo se unieron IBM y American Airlines para crear SABRES, un sistema operativo que fiscalizaba las reservas de vuelos, las transacciones e información de los pasajeros.

Más tarde, Charles Bachman creó un nuevo tipo de BD, esto permitió crear un Estándar en los sistemas de bases de datos gracias a invención de nuevos lenguajes de sistemas de información.

En la década del 70, un informático inglés, llamado Edgar Frank Codd, aclaró el modelo relacional y a su vez anunció una serie de reglas para los sistemas de datos relacionales; a raíz de esto nació la segunda generación de los SGBD.

Gracias al trabajo de Edgar F. Codd, Larry Ellison desarrolló el sistema de software relacional, que actualmente se conoce como Oracle, creando así un sistema de gestión de bases de datos relacional con el nombre de dicha compañía.

En la época de los 80 se creó un lenguaje de consultas para acceder a las bases de datos que permitía realizar consultas para recuperar información de interés de una base de datos y hacer cambios de manera simple; aparte de examinar grandes cantidades de información y deja detallar varios tipos de operaciones a la misma data.

Durante este tiempo SQL comenzó a ser el modelo de la industria; las BD relacionales con su sistema de tablas pudieron competir con las bases jerárquicas y de red.

En los años 90, investigaron las bases de datos orientadas en objetos. Han tenido mucho éxito a la etapa de ejecutar datos complejos en los terrenos donde las bases de datos relacionales no han podido desenvolverse de manera eficaz. Así fueron creadas herramientas como Microsoft Excel y Access.

Así se creó la tercera generación de sistemas gestores de bases de datos.

En esta época también, se incorporaron nuevas expresiones regulares, consultas recursivas y algunas características orientadas a objetos. Además, se creó la posibilidad de que en SQL se pueda utilizar simultáneamente XML, y se determina como importar y guardar datos XML en una base de datos SQL.

Lo extraordinario de esta década es el nacimiento del World Wide Web y gracias a éste es más fácil la consulta a bases de datos.

Actualmente, dominan las bases de datos tres compañías que son IBM, Microsoft y Oracle. En internet google, aunque LINQ también permite crear y manejar bases de datos con facilidad. (GUTIERREZ, S.F)

2.2.7. SQL.

El lenguaje de consulta estructurado (por sus siglas en inglés Structured Query Language) es un lenguaje expresivo de acceso a bases de datos relacionales que permiten especificar muchos tipos de operaciones en ellas. Una de las características de este lenguaje es el manejo del álgebra y el cálculo relacional que permiten generar consultas con el fin de recuperar de forma sencilla la información de necesaria de las bases de datos, así como hacer cambios en ella.

El SQL es un lenguaje de atajo a bases de datos que explota la potencia y flexibilidad de los sistemas relacionales y permite así gran variedad de operaciones. (AREVALO, 2013)

A. Características de Bases de Datos Relacionales.

- Una base de datos relacional está compuesta de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo registro ni nombre.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y una tabla hijo es llevada a cabo por medio de las claves primarias y foráneas.
- Las claves primarias son la parte principal de un registro dentro de una tabla y éstas deben tener integridad de datos.

- Las claves foráneas se ubican en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

B. Ventajas y Desventajas de las Bases de Datos Relacionales.

a. Ventajas

- Proveer herramientas que garanticen evitar la duplicidad de registros.
- Garantizar la integridad referencial, así, al borrar un registro elimina todos los registros relacionados dependientes.
- ayuda la normalización por ser más comprensible y aplicable.

b. Desventajas

- ✓ tienen deficiencias con datos gráficos, CAD, multimedia y sistemas de información geográfica.
- ✓ No se manipulan de forma manejable los bloques de texto como tipo de dato. (AVILA, 2015)

2.2.8. Gestores de Bases de Datos Relacionales.

Existe un software dedicado a presentar con bases de datos relacionales, conocido como Sistema de Gestión de Bases de Datos Relacionales (SGBDR, o RDBMS del inglés Relational Database Management System), también denominados gestores o manejadores de las bases de datos relacionales.

Entre los gestores actuales más populares existen:

- MySQL.

- PostgreSQL.
- Oracle.
- DB2.
- Informix.
- Interbase.
- Firebird.
- Sybase.
- Microsoft SQL Server.

2.2.9. NO SQL.

Cuando hablamos de bases de datos relacionales en nuestra mente acuden los mismos nombres. En la parte comercial tenemos Oracle y Microsoft SQL Server. Por el lado del software libre, tenemos opciones como MySQL o Postgre SQL. Aunque cada una tiene peculiaridades, para los desarrolladores no es difícil elegir entre un sistema y otro. Al final todo son columnas, tablas, claves primarias, y, sobre todo, consultas SQL. La decisión de cuál preferir, se basará en sus características y precio.

Si hablamos de bases de datos no relacionales o NOSQL, la cosa se complica. A día de hoy existen unos 150 sistemas de bases de datos NoSQL. Elegir una es muy difícil, ya que ninguna ha obtenido todavía la fama que sí han conseguido las bases de datos relacionales.

Pero la problemática primordial que encontramos, es que, aunque todas se denominan NoSQL, en realidad hay diferentes tipos. Dependiendo nuestras necesidades, deberemos escoger una u otra.

Aunque hay muchas aproximaciones diferentes para clasificar las NoSQL (Teorema CAP, basándonos en el modelo de datos etc.), en general se piensa que existen cuatro tipos de bases de datos no relacionales: orientadas a documentos, orientadas a columnas, de clave-valor y en grafo. Así que veamos en qué consisten estos sistemas, para que podamos elegir la opción que mejor se acomode a nuestras necesidades. (FMS, 2015)

A. ¿Por qué Aparece NOSQL?

Las bases de datos relacionales no son malas: Precisamente gracias al transcurso de los años, hemos logrado aprender técnicas bastante comunes para normalizarlas en la medida de lo posible, escalarlas según crece la demanda, y utilizarlas como sistema de persistencia para almacenar información desde nuestro lenguaje orientado a objetos preferido. La cuota de uso de software como MySQL, SQLite, PostgreSQL u Oracle, por ubicar estos ejemplos conocidos, es muy alta, encontrándose en la mayor parte de los desarrollos modernos.

Pero llegó la web, los servicios en la nube, el software como servicio y las startups de éxito con millones de usuarios. Y con todo ello se dieron problemas de alta escalabilidad. Si bien los modelos de base de datos relacionales pueden adaptarse para hacerlos escalar incluso en los entornos más difíciles, sí que es cierto que, a menudo, se hacen cada vez menos intuitivo a medida que incrementa la dificultad. Triples y cuádruples JOIN en consultas SQL que inquietan al más experimentado nada más al verlas, a veces poco eficientes, y sistemas de almacenamiento de resultados en cachés para apresurar la resolución de las peticiones y evitar ejecutar cada vez estas pesadas operaciones, son el pan de cada día en muchos de estos proyectos de software.

Los NoSQL atacan este problema proponiendo una estructura de almacenamiento más versátil, aunque sea a costa de inutilizar ciertas funcionalidades como los intercambios que engloban operaciones en más de una colección de datos, o la incapacidad de ejecutar el producto cartesiano de dos tablas (también llamado JOIN) teniendo que usar la desnormalización de datos. (PARAMIO, 2011)

Algunas implementaciones más conocidas que podríamos nombrar como NoSQL son: CouchDB, MongoDB, RavenDB, Neo4j, Cassandra, BigTable, Dynamo, Riak, Hadoop, y otras muchas.

B. Características de no Estructuradas.

- **Consistencia Eventual:** No efectúan mecanismos rígidos de consistencia como los presentes en las bases de datos relacionales, donde la confirmación de un cambio implica la información del mismo a los nodos que lo repliquen. Esta flexibilidad hace que la consistencia se dé, eventualmente, mientras no se hayan modificado los datos durante un periodo de tiempo. Esto es también conocido como BASE (Basically Available Soft-state Eventual Consistency, o coherencia eventual flexible básicamente disponible).
- **Estructura distribuida:** se distribuyen los datos mediante mecanismos de tablas de hash distribuidas.
- **Escalabilidad horizontal:** es la posibilidad de ampliar el rendimiento del sistema simplemente añadiendo más nodos, sin necesidad de realizar ninguna otra operación más que señalar al sistema cuáles son los nodos disponibles. Muchos NoSQL permiten utilizar consultas del tipo Map-Reduce, las cuales pueden ejecutarse en todos los nodos a la vez (cada uno operando sobre una porción de los datos) y reunir los resultados antes de devolverlos.
- **Tolerancia a fallos y Redundancia.**

- **No generan cuellos de botella:** la dificultad de los sistemas SQL, es que deben de transcribir cada sentencia para poder ser ejecutada y, cada sentencia compleja requiere, de un nivel de ejecución más concreto para poderse llevar a cabo, por lo que compone un punto de entrada común, único y conflictivo en base a rendimiento.
- **Solo lo estrictamente necesario:** sistemas simples que no tienen un sistema de consulta complejo ni con capacidad declarativa para en una sola línea realizar una cantidad interna de operaciones exagerada.
- **Estructura dinámica:** significa que los datos no constan de una definición de atributos fija, es decir: Cada registro puede contener una información con distinta forma cada vez, obteniendo así recopilar sólo los atributos que interesen en cada uno de ellos, facilitando el polimorfismo de datos bajo una misma colección de información. También se pueden recolectar estructuras de datos complejas en un documento, como por ejemplo almacenar la información sobre una publicación de un blog (título, cuerpo de texto, autor, etc) junto a las etiquetas y comentarios vertidos sobre el mismo, todo en un único registro. Hacerlo así incrementa la claridad (al tener todos los datos relacionados en un mismo bloque de información) y el rendimiento (no hay que hacer un JOIN para obtener los datos relacionados, pues éstos se encuentran directamente en el mismo documento).

C. Tipos de Bases de Datos no Estructuradas.

- **Key-Valué: clave-valor** es la forma más típica, como un HashMap donde cada elemento está identificado por una llave única, lo que permite la recuperación de la información de manera muy rápida. Normalmente el valor se almacena como un objeto BLOB. De esta forma este tipo de contenido no es muy significativo para la base de datos, solo la clave y el valor que tiene asociado. Son muy eficientes

para lecturas y escrituras, además de que pueden escalar fácilmente particionado los valores de acuerdo a su clave; por ejemplo, aquellos cuya clave está entre 1 y 1000 van a un server, los de 1001 a 2000 a otro server, etc. Varias de estas están basadas en la publicación de Google acerca de su BigTable y de Amazon. Dentro de estas bases de datos también podemos encontrar a BigTable de Google, SimpleDB de Amazon, Cassandra, Hadoop, Riak, Voldemorty MemcacheDB entre otras.

- **Basada en Documentos:** estas almacenan información como documentos (generalmente con una estructura simple como JSON o XML) y con una clave única. Es parecida a las bases de datos Key-value, pero con la diferencia que el valor es un fichero que puede ser entendido. Si el servidor comprende los datos, puede hacer operaciones con ellos. De hecho, muchas 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 poder usar joins. Podemos encontrar a MongoDB y CouchDB entre las más importantes de este tipo.
- **Orientadas a Grafos:** bases de datos que recopilan la información en forma de grafos donde las relaciones entre los nodos son lo más importante. Son útiles para simbolizar información de redes sociales. De hecho, las relaciones también tienen atributos y se pueden concebir consultas directas a relaciones, en vez de a los nodos. Además, al estar almacenadas en esta manera, es mucho más eficiente navegar entre relaciones que en un modelo relacional. Obviamente, este tipo de bases de datos sólo son aprovechadas si

la información en cuestión puede representarse fácilmente como una red. Encontramos a neo4j entre otras.

- **Orientadas a Columnas:** pueden almacenar los valores en columnas en lugar de filas. Con este cambio ganamos velocidad en lectura, ya que, si requieren consultar un número reducido de columnas, es muy rápido hacerlo, pero no es eficiente para realizar escritura. Por ello estas soluciones son usadas en aplicaciones con un índice bajo de escrituras, pero muchas lecturas. Por ejemplo, Cassandra.

2.2.10. Bases de Datos SQL vs Bases de Datos NOSQL.

Cuando usar cada tipo de base de datos:

- ✓ Cuando los datos deben tener consistencia sin dar posibilidad al error es recomendable utilizar una base de datos relacional. SQL.
- ✓ Cuando nuestro presupuesto no nos permite adquirir grandes máquinas y debe destinarse a máquinas de menor rendimiento. NoSQL.
- ✓ Cuando las estructuras de datos que manejamos varían. Usar NoSQL.
- ✓ Análisis de grandes cantidades de datos en modo lectura. Usar NoSQL
- ✓ Captura y procesado de eventos. Usar NoSQL
- ✓ Para Tiendas online con motores de inteligencia complejos. Usar NoSQL. (JAVIER, 2015)

2.2.11. Diferencias entre SQL y NOSQL.

- ✓ SQL permite combinar de manera eficiente diversas tablas para extraer información relacionada, mientras que NoSQL no lo permite o muy limitadamente.
- ✓ NoSQL permite distribuir un gran número de información; mientras que SQL facilita distribuir bases de datos relacionales.
- ✓ SQL permite gestionar los datos conjuntamente con las relaciones existentes entre ellos; en NoSQL no existe este tipo de utilidades.
- ✓ NoSQL permite escalado horizontal sin problemas – por su capacidad de distribución ; mientras que escalar SQL resulta más complicado. (VERGANZA, 2016)

2.2.12. Criterios de uso de Base de Datos SQL.

No existe una respuesta concreta, porque se supone que cualquier cosa que se deba guardar, debe ser en una base de datos relacional. Sin embargo, los programadores pueden sugerir que en gaming o desarrollos de animación, no es necesario.

- ✓ su uso es más adaptado y los perfiles que los conocen son mayoritarios y más baratos.
- ✓ Por tener más tiempo en el mercado, dichas herramientas tienen un mayor soporte y mejores suites de productos y add-ons para gestionar estas bases de datos.
- ✓ La atomicidad de las operaciones en estas bases de datos. Esto es, que en estas bases de datos o se hacen las operaciones enteras o no se hacen utilizando la famosa técnica del rollback.

- ✓ Los datos deben efectuar integridad tanto en tipo de dato como en compatibilidad.

En la gran mayoría de opiniones, una base de datos relacional puede ser usada los siguientes ámbitos:

- En la Educación: para estructurar información, y aportar conocimiento lógico al estudiante.
- En el Desarrollo web: para conservar jerarquía de datos, siempre y cuando la capacidad de concurrencia, almacenamiento y mantenimiento no sean de considerable problema y la información sea consistente.
- En los Negocios: la inteligencia y análisis de negocios, son temas que demandan el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- En las Empresas: porque tanto el software a la medida y el software empresarial, poseen la característica de mantener información con estructura consistente.

2.2.13. Criterios de uso de Base de Datos NOSQL.

- ✓ La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas.
- ✓ son bases de datos mucho más abiertas y flexibles. Permiten adaptarse a necesidades de proyectos mucho más fácilmente que los modelos de Entidad Relación.
- ✓ Se pueden hacer cambios de esquemas sin necesidad de parar bases de datos.

- ✓ Escalabilidad horizontal: tienen la capacidad de crecer en número de máquinas, en lugar de tener que adquirir grandes máquinas.
- ✓ Se pueden ejecutar en máquinas con menores recursos.
- ✓ Optimización de consultas en base de datos para grandes cantidades de datos.

prácticamente se utilizarían en:

- Redes sociales: casi obligatorio por el tipo de información que crece abismalmente.
- Desarrollo Web: debido a la baja uniformidad de la información que se halla en Internet; aun cuando también puede emplearse SQL.
- Desarrollo Móvil: debido a la tendencia que está en crecimiento- de Bring Your Own Device (trae tu propio dispositivo).
- BigData: debido a la administración de grandes montos de información y su evidente heterogeneidad.
- Cloud (XaaS): “Everything as a service”; NoSQL puede adaptarse casi a cualquier necesidad del cliente, y sus particularidades. (JURADO, S.F).

2.2.14. Ejemplos de bases de datos NOSQL.

Veamos a continuación algunas tipas de bases NoSQL más utilizadas actualmente.

1.- Cassandra.

Es una base de datos creada por Apache del tipo clave–valor. Dispone de un lenguaje propio para realizar consultas CQL (Cassandra Query Language). es una aplicación Java por ello puede circular en cualquier plataforma que cuente con la JVM (máquina virtual java).

2.- Redis.

Es una base de datos creada por Salvatore Sanfilippo y Pieter Noordhuis y está apoyado por VMWare. Se trata de una base de datos del tipo clave–valor. Se puede visionar como un array gigante en memoria para almacenar datos, datos que pueden ser cadenas, hashes, conjuntos de datos o listas. Tiene la ventaja de que sus operaciones son persistentes y atómicas. Por ponerle un contra, Redis no accede a realizar consultas, sólo se puede insertar y obtener datos, además de las operaciones comunes sobre conjuntos (diferencia, unión e inserción). Creado en ANSI C, por lo tanto, funciona y es compatible sin dificultades en sistemas Unix, Linux y sus derivados, Solaris, OS/X sin embargo no existe soporte oficial para Windows.

3.- MongoDB.

Se trata de una base de datos creada por 10gen orientada a documentos, de esquema libre, es decir, que cada entrada puede tener un esquema de datos distinto y que nada tenga que ver con los otros registros almacenados. Es bastante rápido a la hora de ejecutar sus operaciones ya que está escrito en lenguaje C++. Para el almacenamiento de la información, maneja un sistema propio de documentos conocido como BSON, que es una evolución del conocido JSON pero con la peculiaridad de que puedan

almacenar datos binarios. En menor tiempo, MongoDB se ha convertido en una de las bases de datos NOSQL favoritas por los desarrolladores.

4.- CouchDB.

Es un sistema creado por Apache y escrito en lenguaje Erlang que funciona en la mayoría de sistemas POSIX, incluyendo GNU/LINUX y OSX, pero no así en sistemas Windows. Una de las características más significativas es el que usan de Restfull HTTP API como interfaz y JavaScript como principal lenguaje de interacción. Para almacenar los datos se utilizan archivos JSON. Permite la creación de vistas, que son mecanismos que permiten combinar documentos para retornar valores de varios documentos, es decir, CouchDB permite la realización de las operaciones JOIN típicas de SQL.

2.2.15. Compañías que usan NOSQL.

Son muchas las empresas que hacen uso de este tipo de bases de datos como son:

- Cassandra: Facebook, Twitter...
- HBase: Yahoo, Adobe...
- Redis: Flickr, Instagram, Github...
- Neo4j: Infojobs...
- MongoDB: FourSquare, SourceForge, CERN...

2.2.16. Big data.

Debido al avance actual en tecnologías de información, las organizaciones se han tenido que enfrentar a nuevos retos que les permitan

descubrir, analizar y entender más allá de lo que sus herramientas tradicionales reportan sobre su información, al mismo tiempo que durante los últimos años el gran crecimiento de aplicaciones disponibles en internet (redes sociales, geo-referenciamiento etc.)

¿Qué es Big Data y porqué se ha vuelto tan importante?

Podríamos referirnos en términos generales, como la tendencia en el avance tecnológico que ha abierto las puertas a un nuevo enfoque en toma de decisiones y entendimiento, la cual es utilizada para describir grandes cantidades de datos, que tomaría demasiado tiempo y sería muy costoso cargarlos a un base de datos relacional para su análisis, De esta forma el concepto de Big Data emplea para toda información que no puede ser procesada o analizada utilizando procesos o herramientas tradicionales.

Además de las gran cantidad de información, hay una gran variedad de datos que pueden ser representados de muchas maneras en todo el mundo, como los de audio, video, dispositivos móviles, sistemas GPS, muchos sensores digitales en equipos industriales, medidores eléctricos, automóviles, veletas, anemómetros, etc., estos pueden comunicar y medir el posicionamiento, vibración, movimiento, humedad, temperatura y hasta los cambios químicos que sufre el aire, de tal forma que las aplicaciones que analizan estos datos requieren que la velocidad de respuesta sea lo más rápida para conseguir la información correcta en el momento preciso. Estas son las características principales de una oportunidad para Big Data. (FRAGOSO, 2012)

2.3. Definiciones conceptuales.

1. **Big data:** la disponibilidad de grandes cantidades de información en formatos estructurados y desestructurados en tiempo real. En ese

orden de ideas, el Volumen, la Variedad y la Velocidad de los datos son dimensiones clave del concepto big data. (salvador.2014).

2. **XML:** no es, como su nombre podría sugerir, un lenguaje de marcado. XML es una meta-lenguaje que nos permite poder definir lenguajes de marcado apropiados a usos determinados. (Romero.2016).
3. **SQL:** El lenguaje estructurado de consultas SQL, ayuda a la creación y mantenimiento de la base de datos relacional y la gestión de los datos dentro de la base de datos. (Oppel.2009).
4. **Base de datos:** colección de datos organizada en un formato estructurado que es definido como metadatos que describe esa estructura. Puede pensar en los metadatos como información sobre los datos almacenados, que define cómo se almacenan éstos en una base de datos. (Oppel.2009).
5. **Json:** es un formato ligero de intercambio de datos, independiente del lenguaje de programación. Tiene forma de texto plano, de simple lectura, escritura y generación. Y además ocupa menos espacio que el formato XML (Jaramillo.2011).
6. **Joins:** es una operación que busca hacer coincidir las filas en una tabla, con las filas de manera tal que las columnas de ambas tablas puedan ser colocadas lado a lado en los resultados de la consulta como si éstos vinieran de una sola tabla. (Oppel.2009).
7. **Bigtable:** es un mapa continuo y ordenado. Cada cadena en el mapa consta de una fila, las columnas (varios tipos) y un valor de marca de tiempo que se utiliza para la indexación.

(<https://cloud.google.com/bigtable/2016>)

8. **Base de datos relacional:** Los datos se muestran en forma de tablas y relaciones. Este es el modelo que se comenta en el presente documento. De hecho, es el claramente más popular. (Sánchez 2004).

9. **BASE:** Disponibilidad básica., Prioridad de disponibilidad de datos, Se prioriza la propagación de datos, delegando el control de inconsistencias a elementos externos.

Consistencia eventual. Se asume que inconsistencias temporales progresen a un estado final estable.

(<https://mimundobinario.wordpress.com/2016/03/09/event-sourcing/2016>)

10. **ACID:** (Atomicity Consistency Insolation Durability), ACID es un acrónimo en inglés que utiliza en el ámbito de las bases de datos. A menudo se utilizan expresiones como "esta base de datos o este sistema cumple las propiedades ACID".
(<http://latecladeescape.com/h/2015/07/acid>)

CAPÍTULO III

3. Metodología de la Investigación

3.1. Tipo de investigación (Referencial)

Expresa sus objetivos como descripciones y relaciones entre variables” La investigación es cualitativa, en sus múltiples modalidades: investigación de campo, investigación participativa, etnográfica, estudio de casos, etc., “tienen como característica el referirse a sucesos complejos que tratan de ser descritos en su totalidad, en su medio natural. No hay consecuentemente, una abstracción de variables o propiedades para analizarlas a través de técnicas estadísticas adecuadas para su descripción y la determinación de correlaciones.” Los investigadores cualitativos estudian la realidad en su forma natural, pretendiendo sacar sentido de, o desentrañar, los fenómenos de acuerdo con los significados que se dan para las personas implicadas. Este tipo de investigación involucra utilizar y recoger una gran diversidad de materiales que describen la rutina y las situaciones problemáticas y los significados en la vida de las personas. (Gregorio Rodríguez Gómez y otros, METODOLOGÍA DE LA INVESTIGACIÓN CUALITATIVA, 1996:72) Enfoque

El enfoque es Cualitativo por que se describen las cualidades de las bases de datos SQL y las NOSQL.

3.1.1. Alcance o nivel

El alcance es Exploratorio por qué este tema aún no ha sido abordado o conocido y, se hace una comparación entre estos tipos de bases de datos relacionales y no relacionales.

3.1.2. Diseño

El diseño es Explicativo por que se hacen las comparaciones entre estos tipos de bases de datos para poder escoger el que se ajuste mejor a nuestra empresa.

3.2. Técnicas e instrumentos de recolección de datos

La recolección de datos será por medio de bibliografía para dar un panorama más amplio para el uso de bases de datos relacionales y no relacionales.

3.3. Técnicas para el procesamiento y análisis de la información

Se usará la técnica bibliográfica para saber las preferencias de uso de bases de datos y si se podría implementar una base de datos NOSQL.

CAPÍTULO IV

4.1. Procesamiento de datos

Tabla 1. Diferencias entre las bases de datos relacionales y no relacionales

Base de datos Relacional	Base de datos No relacional
Escalabilidad Baja	Escalabilidad Alta
Rendimiento bajo	Rendimiento alto
La fiabilidad es alta	La fiabilidad es baja
Utiliza propiedades ACID	Utiliza propiedades BASE
Implementación costosa	Costo de implementación
Alta seguridad	Muy baja seguridad
Procesamiento de datos lento	Procesamiento de datos veloz

Fuente: <http://www.kasperu.com/courses/BG011/II/default.htm>

Detalle

Si bien NoSql, no pretende erradicar ni suplantar al SQL y a las bases de datos relacionales, sino pretende ser una alternativa dentro del desarrollo del software donde las condiciones sean requeridas principalmente cuando se manejan millones y millones de datos y el procesamiento y escalabilidad sean

puntos críticos.

Aunque sabemos que las bases de datos son sistemas de almacenamiento de información, estas también cuentan con un límite de registros o datos máximo, no porque ya no se puedan almacenar registros sino porque empiezan a crecer los tiempos de espera.

Tabla 2. Comparaciones de bases de datos relacionales y no relacionales

Descripción	No relacional	Relacionales
Estructura de datos	Es una estructura flexible, no es necesario definir una estructura de datos	Asume una estructura bien definida de datos tienen que ser uniformes y las propiedades de estos datos pueden definirse por
Relaciones entre tablas	No existe relación entre	Tiene que estar un muy bien
	Colección, no obstante, puede depender del modelo de datos.	Establecidas y ser referenciadas de forma sistemática.
Transaccionalidad	Se pierde integridad en las transacciones	Utiliza ACID
Consultas e índices	Disminuye el uso de indexación y el poder de las consultas	

Fuente: investigación propia

Detalle

Mostraremos las diferencias entre las bases de datos relacionales y no relacionales en la siguiente tabla.

Tabla 3. Características de ACID y BASE

ACID	BASE
❖ Aislamiento	❖ Disponibilidad
❖ Consistencia fuerte	❖ Consistencia débil
❖ Poca disponibilidad	❖ Intuitivo
❖ Evolución difícil (Escalabilidad)	❖ Evolución más fácil
❖ Transacciones anidadas	❖ Respuestas más rápidas
❖ Requiere "commit"	❖ Mejor esfuerzo

Fuente: <http://www.dosideas.com/noticias/base-de-datos/973-acid-en-las-bases-de-datos.html>

Detalle

Entre las características que posee NoSQL se encuentra que no presentan esquemas, tienen fácil soporte de replicación, API simple, eventualmente consistente (conocido como BASE, y contrario al concepto de ACID) y contienen enormes cantidades de datos.

Para una mayor profundidad y calidad sobre las bases de datos NoSQL que hemos mencionado al principio de este trabajo realizado destacamos las comparaciones.

4.2. Criterios de uso de base de datos SQL Y base de datos NOSQL

Una base de datos relacional puede ser usada en los siguientes ámbitos:

- Educación: para aportar conocimiento lógico al estudiante y para estructurar información.

- Desarrollos web: para poseer jerarquía de datos, siempre que la capacidad de concurrencia, mantenimiento y almacenamiento no sean de dificultad considerable y la información sea consistente.
- Negocios: análisis de negocios e inteligencia de negocios, requieren el uso de SQL para facilitar el consumo de la información y la identificación de patrones en los datos.
- Empresarial: porque el software a la medida y el software empresarial, tienen la característica de mantener información con estructura consistente.

Básicamente se utilizan las bases de datos no relacionales en:

- Redes sociales: casi obligatorio.
- Desarrollo Web: debido a la poca semejanza de la información que se encuentra en Internet; aun cuando también puede emplearse SQL.
- Desarrollo Móvil: debido a la tendencia de crecimiento- de Bring Your Own Device. (utilizar dispositivos portátiles personales para realizar tareas).
- BigData: debido a la administración de grandísimas cantidades de información y su evidente heterogeneidad.
- Cloud (XaaS): “Everything as a service”; NoSQL puede adaptarse casi a cualquier necesidad del cliente, y sus particularidades.

Tabla 4. Diferencias ente SQL y NOSQL

Base de datos SQL	Base de datos NoSQL
Escalabilidad Baja	Escalabilidad Alta
Rendimiento bajo	Rendimiento alto
La fiabilidad es alta	La fiabilidad es baja
Utiliza propiedades ACID	Utiliza propiedades BASE
Implementación costosa	Costo de implementación moderados
Alta seguridad	Muy baja seguridad
Procesamiento de datos lento	Procesamiento de datos veloz

Detalle

Código Abierto vs Cerrado

Como su nombre lo indica, abierto permite la utilización del código fuente por parte de los usuarios para su modificación creando grupos que ayudan a compartirlo y a mejorar el código y puede ser distribuido pagado o gratuitamente. No puede ser cobrado y la licencia debe ser libre; el código cerrado (propietario) puede ser cobrado, pero no permite el acceso al código fuente. como consecuencia del código abierto en NOSQL, existe una gran oferta de propuestas en el mercado.

Ejemplo de base de datos propietarios: Big Table de Google, Dynamo DB y simplemente DB de Amazon.

Ejemplos de base de datos libres: Cassandra de Facebook , Couch DB de apache , Redis , Neo4j, Mongo DB.

CONCLUSIONES

Con ayuda de la información obtenida producto de la investigación, es posible saber qué tipo de base de datos utilizar y en qué casos, ya que tenemos un panorama mayor producto de las definiciones e información captada, así mismo con la ayuda del proyecto de investigación podremos saber en qué casos usar cada una de ella y cual se adapte mejor a nuestro trabajo.

Las bases de datos relacionales son más utilizadas a nivel de las empresas ya que, la mayoría de empresas prefieren utilizar una base de datos conocida, a diferencia de las bases de datos NOSQL que recién ingresan al mercado y pocos conocen.

Las bases de datos NOSQL son más utilizadas en las redes sociales, páginas web de empresas que realizan ventas en línea, y otros que manejen gran volumen de información,

Las bases de datos relacionales son más utilizadas en empresas como: colegios, instituciones públicas, pequeñas empresas y grandes que no manejen gran cantidad de data, por ello las conclusiones de este proyecto serian, que dependiendo el tipo de empresa que se tenga, se elegiría la base de datos que se acomode a ello.

RECOMENDACIONES

Las bases de datos NOSQL son más recomendables para proyectos grandes, ya que, al tener gran cantidad de información, las consultas en SQL son lentas por el hecho de tener que obtener información de muchas tablas, así mismo para el uso de empresas que desean obtener información más normalizada se recomienda usar base de datos SQL.

Al realizar esta clase de proyectos debemos saber que se debe de realizar un análisis bastante minucioso, es vital determinar cada uno de los requisitos y características para, que de esta manera se pueda cubrir con las necesidades requeridas para de esta manera evitar gastos innecesarios.

En el caso de una base de datos relacional, se debe desarrollar un buen diseño para que la información no contenga datos redundantes.

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.

5. BIBLIOGRAFÍA

Internetría Consultoría Digital. (8 de Mayo de 2016). Recuperado el 15 de Noviembre de 2016, de <http://www.internetria.com/blog/2016/05/08/nosql/>

SQL Thoughts from IngeniousSQL. (21 de Febrero de 2016). Recuperado el 10 de Noviembre de 2016, de <http://www.ingenioussql.com/tag/key-value-store/>

BARRAGAN CHARRY, A. M. (2016). DSpace Repository. Recuperado el 10 de Diciembre de 2016, de <http://repository.ucatolica.edu.co/xmlui/handle/10983/690?show=full>

BASES DE DATOS NOSQL: LLEGARON PARA QUEDARSE.

Domenjoud, M. (12 de Julio de 2016). logo Octo Technology. Recuperado el 10 de Noviembre de 2016, de <http://blog.octo.com/en/graph-databases-an-overview/>

Base de datos. Consultado el 20 julio 2016, de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

Base de datos jerárquica. Consultado 20 de julio 2016, de <http://www.tecnologias-informacion.com/basesdedatos.html>

Bases de datos en red. Consultado 20 de julio 2016, de <http://www.dataprix.com/262-bases-datos-red>

Base de datos relacional. Consultado 20 de julio 2016, de <https://elbaultdelprogramador.com/disenio-de-bases-de-datos-i-conceptos/>

Base de datos relacional. Consultado 20 de julio 2016, de <https://elbaultdelprogramador.com/disenio-de-bases-de-datos-i-conceptos/>

Bases de datos relacionales. (Miércoles 8 de junio del 2011). consultado el 20 de julio del 2016, de <http://sergioyrafael-informatica.blogspot.pe/2011/06/ventajas-y-desventajas.html>

Base de datos NOSQL elige la que mejor se adapte a tus necesidades. Consultado 20 de julio del 2016, de

<http://www.genbetadev.com/bases-de-datos/bases-de-datos-nosql-elige-la-opcion-que-mejor-se-adapte-a-tus-necesidades>

Bases de datos SQL que son y tipos que nos podemos encontrar. Consultado 20 de julio 2016, de <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

Conceptos básicos de SQL. Consultado 20 de julio 2016, de http://geotalleres.readthedocs.io/es/latest/conceptos-sql/conceptos_sql.html

El concepto de NOSQL o como almacenar tus datos en una base de datos no relacional. Consultado 20 de julio 2016, de <http://www.genbetadev.com/bases-de-datos/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional>

Historia de las bases de datos. Consultado 20 de julio 2016, de <http://dryvalleycomputer.com/index.php/bases-de-datos/introduccion/45-historia-de-las-bases-de-datos>

Introducción a las bases de datos. Consultado el 20 de julio 2016, de http://www.aulaclie.es/sqlserver/b_1_1_4.htm

NOSQL y SQL diferencias y cuando elegir cada una. Consultado 20 de julio 2016, de <http://blog.pandorafms.org/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>

Que es big data. (18 junio 2012). consultado 20 de julio 2016, de <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>

Que es un gestor y cuáles son sus tipos. Consultado 20 de julio 2016, de <http://brenyova.blogspot.pe/>

Que es no SQL. Consultado 20 de julio 2016, de <http://codecriticon.com/que-es-nosql/>

SQL vs NOSQL cual debo usar. Consultado 20 de julio 2016, de https://www.facilcloud.com/noticias/es_ES/sql-vs-nosql-which-one-should-i-use/

SQL vs NOSQL cual debo usar. Consultado 20 de julio 2016, de https://www.facilcloud.com/noticias/es_ES/sql-vs-nosql-which-one-should-i-use/

Tipos y función de los gestores de base de datos. Consultado 20 de julio 2016, de <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/406547/Tipos-y-funci-n-de-los-gestores-de-bases-de-datos>

Candia Condori , M. L. (2016). NoSQL en la NUBE. revistasbolivianas.

Castro Romero, A., González Sanabria, J. S., & Callejas Cuervo, M. (2016). Utilidad y funcionamiento de las bases de datos NoSQL. 21.

Díaz Sepúlveda, W. (2016).

ANEXO

MATRIZ DE CONSISTENCIA

PROBLEMA	OBJETIVOS	HIPOTESIS	VARIABLES	INDICADORES	Metodología
P.E.1.en qué casos se recomienda usar bases de datos relacionales.	O.E.1.Determinar los casos en los que se recomienda usar bases de datos relacionales.	H.E.1.Conociendo los tipos de bases de datos relacional podremos determinar sus criterios de uso.	Criterios de uso	<ul style="list-style-type: none"> ✓ La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas. ✓ Son bases de datos más flexibles y abiertas. Se adaptan fácilmente a las necesidades de los proyectos en comparación a los modelos de Entidad Relación. ✓ Sin parar a las bases de datos, es posible hacer cambios de los esquemas. ✓ Escalabilidad horizontal: se consigue crecer en número de máquinas, en vez de tener que residir en máquinas grandes. ✓ Es posible ejecutar en máquinas con pocas características o recursos. ✓ Se Optimizan las consultas en las bases de datos para grandes cantidades de datos. 	El presente trabajo de investigación nos permitirá obtener un mayor conocimiento acerca de los criterios de uso de las bases de datos relacionales y las bases de datos NOSQL, para así poder mejorar los procesos de las empresas que tienen gran volumen de información

<p>P.E.2. en qué casos en los que se recomienda usar bases de datos NOSQL.</p>	<p>O.E.2.Determinar los casos en los que se recomienda usar bases de datos NOSQL.</p>	<p>H.E.2. conociendo los tipos de bases de datos NOSQL podremos determinar los criterios de uso.</p>	<p>Implementación exitosa de la base de datos.</p>	<ul style="list-style-type: none"> ✓ su uso es más adaptado y los perfiles que los conocen son mayores y más baratos. ✓ Al tener más tiempo en el mercado, estas herramientas poseen más soporte y mejores suites de productos y add-ons para la gestión de las bases de datos. ✓ La atomicidad, es decir, que en estas bases de datos o se hace la operación completa o no se hace usando la técnica del rollback. ✓ La integridad de datos se cumple, tanto en tipo de dato como en compatibilidad. 	
--	---	--	--	---	--