



UNIVERSIDAD
PRIVADA
DEL NORTE

FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“ESTUDIO COMPARATIVO ENTRE LAS BASES
DE DATOS RELACIONAL Y NO RELACIONALES:
UNA REVISIÓN DE LA LITERATURA CIENTÍFICA”

Trabajo de investigación para optar el grado de:

Bachiller en Ingeniería de Sistemas Computacionales

Autores:

Jairo Giordani Caqui Vargas

Jean Pierre Pareja Limaco

Asesor:

Ing. Manuel Ricardo Mondragón Vilela

Lima - Perú

2018

DEDICATORIA

A nuestras familias.

AGRADECIMIENTO

A las personas que siempre nos apoyaron.

Tabla de contenido

DEDICATORIA	2
AGRADECIMIENTO	3
ÍNDICE DE TABLAS	5
ÍNDICE DE FIGURAS	6
RESUMEN	7
CAPÍTULO I. INTRODUCCIÓN	8
CAPÍTULO II. METODOLOGÍA	24
CAPÍTULO III. RESULTADOS	31
CAPÍTULO IV. CONCLUSIONES	32
REFERENCIAS	33

ÍNDICE DE TABLAS

Tabla 1. Principales características y diferencias entre las bases de datos SQL y NOSQL	18
Tabla 2. Resumen de la Estrategia de Búsqueda	24
Tabla 3. Cantidad de artículos encontrados en primera fase de búsqueda	26
Tabla 4. Cantidad de artículos encontrados aplicando filtros de selección	26
Tabla 5. Resultados de artículos encontrados para la revisión	27
Tabla 6. Formato de tabla de extracción de datos.....	27
Tabla 7. Distribución de artículos por fuente.....	28
Tabla 8. Distribución de base datos por articulo y revistas	29
Tabla 9. Distribución de termino clave por base de datos	29
Tabla 10. Distribución de artículos encontrados por países.....	29
Tabla 11. Distribución de artículos incluido y excluidos según base de datos	30
Tabla 12. Distribución de artículos incluido y excluidos según base de datos.....	30

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de filas de empleados en una BD NoSQL orientada a columnas	17
Figura 2. Ejemplo de filas empleados en un BD relacional orientada a filas	17
Figura 3. Representación de datos en una BD NoSQL de tipo grafo.....	18
Figura 4. Comparación de términos entre una BD relacional y una BD orientada a documentos (MongoDB).....	21
Figura 5. Formas generales de las sentencias básicas de definición del esquema en SQL y en MongoDB	21
Figura 6. Ejemplos de las sentencias de definición del esquema en SQL y en MongoDB.....	22
Figura 7. Formas generales de las sentencias de inserción, actualización y borrado de datos en SQL y en MongoDB	22
Figura 8. Ejemplos de las sentencias de inserción, actualización y borrado de datos en SQL y en MongoDB	23
Figura 9. Ejemplos de las consultas en SQL y en MongoDB	23
Figura 10. Gráfico de artículos por fuente	28

RESUMEN

Los sistemas de información actualmente deben procesar de forma eficiente grandes volúmenes de información para una correcta toma de decisiones. Existen diferentes modelos de base de datos que realizan estas gestiones, los más utilizados son el modelo relacional (SQL) y no relacional (NOSQL). Cada uno de estos modelos tienen características propias que definen su uso.

El objetivo principal de este estudio es poder comprender las características principales y establecer las diferencias entre estas bases de datos. A través de la recopilación de artículos de investigación se ha podido obtener resultados que respondan al objetivo del tema de investigación.

PALABRAS CLAVE: Bases de datos, SQL, NOSQL.

CAPÍTULO I. INTRODUCCIÓN

Una base de datos empresarial nos permite almacenar grandes cantidades de información de forma organizada para poder en un futuro encontrarla y utilizarla fácilmente.

Desde el momento en que los humanos han requerido almacenar datos que lleven a la obtención de información oportuna y fiable también se ha buscado la manera más eficiente de manipular dichos datos. Con el tiempo la creciente cantidad de información que se intenta manejar supera por mucho la infraestructura existente, esto es, se ha llegado a la necesidad de buscar soluciones que permitan almacenar cada vez más información incluso de la que se podría imaginar. (Antaño, Castro y Valencia, 2014)

Las bases de datos relacionales están presentando algunos inconvenientes, y como sucede con la tecnología; continuos cambios en las sociedades impulsan la aparición de modelos que satisfagan nuevas necesidades, uno de estos es NoSQL, que define un conjunto de tecnologías que distan de lo planteado por los gestores de bases relacionales. En la actualidad es decisivo para el éxito de las empresas la capacidad de procesar una gran cantidad de datos de una amplia gama de fuentes en cualquier lugar y en cualquier momento de manera eficiente. El análisis de datos se convierte en una estrategia clave para la mayoría de las grandes organizaciones para lograr una ventaja competitiva. Por tanto, surgen nuevas cuestiones a ser consideradas a la hora de almacenar y consultar cantidades masivas de datos que, en general, las bases de datos relacionales tradicionales no pueden abarcar. Estas cuestiones incluyen desde la capacidad de distribuir y escalar el procesamiento o el almacenamiento físico, hasta la posibilidad de utilizar esquemas o tipos de datos no usuales. (Cazar, Wilson, Gaona y Javier, 2018).

Hoy en día empieza a haber una tendencia por la utilización de Bases de Datos No SQL. Pareciera que la moda en estos momentos es migrar a bases de datos NoSQL, pero debemos tener en cuenta antes de tomar una decisión las características de la información

que se desea almacenar para elegir entre las bases de datos NoSQL o relacional. Un punto de partida es necesidad de gestionar grandes volúmenes de información generada por la alta demanda producida por la evolución de las aplicaciones web; esto ha llevado a una nueva tendencia en el manejo de bases de datos conocida como NoSQL, que ofrece una mayor libertad en la definición de esquemas, ya que la responsabilidad recae sobre la aplicación y no sobre el motor de bases de datos. Adicionalmente fortalece dos requerimientos importantes en una aplicación como son la escalabilidad y la disponibilidad. Esto se puede evidenciar en aplicaciones como las redes sociales, que presentan gran dinamismo y rendimiento. De acuerdo a su forma de almacenar la información, se clasifican en: llave valor, familias de columnas, almacenes de documentos y de grafos. (Herrera y Valenzuela, 2016).

ANTECEDENTES

La primera vez que se escuchó este término fue en el año 1963, en un simposio en California, refiriéndose a un conjunto de información que se encuentra agrupada o estructurada. Desde entonces, han evolucionado mucho, tanto en capacidad como en la información contenida, especialmente en el caso de las de carácter empresarial. (Universia, 2017)

En 1970 se propusieron por primera vez las bases de datos relacionales y las teorías subyacentes, entre las que se destaca el modelo de base de datos relacional, que implicó un cambio radical en el manejo de la información apoyándose en operaciones de conjuntos que combinan tablas de datos separadas (o relaciones) para producir un conjunto de respuestas. Las consultas se especifican utilizando el lenguaje de consulta estructurado SQL (por las siglas en inglés de Structured Query Language), soportado en el álgebra relacional, y que permite a un usuario expresar su consulta en forma declarativa, sin ningún tipo de instrucciones detalladas de programación. (Castro, González y Callejas, 2012).

En 1998 fue acuñado el término NoSQL, se refería a una base de datos relacional de código abierto que no usaba un lenguaje de consultas SQL (Structured Query Language).

Hasta 2009 estas cinco letras cayeron en el olvido, pero fue Johan Oskarsson, entonces empleado de Last.fm, quien organizó un evento para tratar las bases de datos distribuidas de código abierto no relacionales, llamándolas “NOSQL”, Not-Only SQL. (Ecured) que en principio estaban destinados al manejo de grandes volúmenes.

A continuación, se citan fragmentos de autores que tratan sobre el tema de las bases de datos SQL y NoSql:

Según Moreno, Quintero y Rueda (2016) nos informan sobre la creciente y enorme cantidad de datos, del orden de exabytes, generados por las aplicaciones empresariales actuales han originado conjuntos masivos de estos. Los sistemas de gestión de bases de datos (SGBD) NoSQL han surgido como una alternativa a los SGBD relacionales para la gestión de estos conjuntos.

Sadalage (2013) nos indica que las bases de datos SQL parecen no ser suficientes para tratar con tan amplio universo de problemas que vienen ligados a la aparición de Big Data. Esto generó la llegada de las tecnologías para el almacenamiento de datos englobadas dentro del concepto de NoSQL.

Según Jaramillo y Londoño (2014) afirman que se puede estimar que aquellas bases de datos con esquemas más rígidos son adecuadas para manejar estructuras de datos con información descriptiva, sin demasiadas relaciones entre ellas. No obstante, dada la cantidad de datos utilizada, ésta resulta insuficiente para realizar una afirmación al respecto.

Hoy se utilizan los dos tipos de bases de datos en las empresas, pero no para realizar las mismas funciones, ni van dirigidas al mismo tipo de público. En función de las necesidades de cada uno, unas serán más útiles que otras.

La diferencia conceptual entre SQL y NoSQL es que resuelven escenarios completamente diferentes y excluyentes el uno del otro; ya que para lo que resulta ideal SQL, no lo es NoSQL y al revés.

JUSTIFICACIÓN

Actualmente el criterio para escoger una base de datos en las empresas es según necesidad o expertis del usuario, pero ese criterio de elección no es del todo certero, con el tiempo las organizaciones generan más volumen de información y esta los supera, esto provoca que el rendimiento disminuya. Además, si nos guiáramos de la información que existe en la web (para la elección de una base de datos), nos encontraríamos con todo tipo de información no objetiva.

Muchos usuarios son partidarios del SQL por su estabilidad conceptual a través de los años y otros los son del NOSQL por tendencia y practicidad.

Es por ello que se ha visto la necesidad de analizar y documentar de forma clara y concisa las diferencias que existen entre estos dos modelos y así elegir el más conveniente según necesidad en el ámbito empresarial. Para la elaboración de este estudio o análisis se ha recopilado artículos de investigación que tratan sobre estos temas (bases de datos SQL y NOSQL) de forma que justifiquen la calidad y nos den un criterio para la elección del mismo.

FORMULACIÓN DEL PROBLEMA

De acuerdo a la búsqueda realizada evidenciamos que no existe mucha información referente o información descriptiva acerca de las diferencias existentes entre las bases de datos SQL y NOSQL.

Al no tener el conocimiento necesario para determinar qué base de datos sería la adecuada para utilizar en nuestras aplicaciones a desarrollar no podemos elegir entre ambas. Es por ello nuestro problema de estudio, el determinar cuáles son las diferencias existentes entre las bases de datos SQL y NOSQL.

OBJETIVOS

La presente revisión sistemática tiene como objetivo principal dar a conocer estas diferencias entre las bases de datos SQL y NOSQL de forma que se pueda elegir e implementar (de ser necesario) en los proyectos que implican el almacenamiento y gestión de la información.

DEFINICIONES

Base de datos

Para definir “bases de datos” es importante tener claro qué es un dato e información debido a que estos elementos son fundamentales para el desarrollo de las bases de datos.

Según Juárez (2006):

- **Dato:** es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos, este es la unidad mínima de información. Un dato dentro de una base de datos responde a la función (objeto, atributo, valor).
- **Información:** es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos.

En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para el sistema. Se trata de la información relevante que almacena y gestiona un sistema de información. Ejemplos de datos son: Sánchez, 12764569F, Calle Mayo 5, etc. Desde el inicio de la informática se ha reconocido al dato como elemento fundamental de trabajo en un ordenador. (Sanchez, 2013).

Los Datos por sí solos no significan nada más que representaciones simbólicas, siendo por ello llamados Datos Aislados cuando no tienen correlación alguna o no llevan un Orden específico, de lo contrario se tratarían de unidades propias de una Información, considerándose a esta última como un conjunto de Datos debidamente procesados y contextualizados.

Una base de datos es un almacén de datos relacionados con diferentes modos de organización. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al usuario. Y que almacena datos con un propósito específico. Con la palabra “datos” se hace referencia a hechos conocidos que pueden registrarse. (Gutierrez, s.f).

Una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso, es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción han de ser únicas estando almacenadas junto a los mismos. (Gómez, 2007)

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. (Juárez, 2006)

Sistema Gestor Base de Datos (SGBD)

Definimos un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD, (DB DataBase).

Base de datos Relacional (SQL)

Según la web de Amazon Web Services (AWS), una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos. Cada columna de una tabla guarda un determinado tipo de datos y un campo almacena el valor real de un atributo. Las filas de la tabla representan una recopilación de valores relacionados de un objeto o entidad. Cada fila de una tabla podría marcarse con un identificador único denominado clave principal, mientras que filas de varias tablas pueden relacionarse con claves extranjeras. Se puede obtener acceso a estos datos de muchas formas distintas sin reorganizar las propias tablas de la base de datos.

Según el modelo relacional (desde que Codd lo enunció) el elemento fundamental es lo que se conoce como relación, aunque más habitualmente se le llama tabla (o también array o matriz). Codd definió las relaciones utilizando un lenguaje matemático, pero se pueden asociar a la idea de tabla (de filas y columnas) ya que es más fácil de entender. No hay que confundir la idea de relación según el modelo de Codd, con lo que significa una relación en el modelo Entidad/Relación de Chen. No tienen nada que ver

Las relaciones constan de:

- **Atributos.** Referido a cada propiedad de los datos que se almacenan en la relación (nombre, DNI...).
- **Tuplas.** Referido a cada elemento de la relación. Por ejemplo, si una relación almacena personas, una tupla representaría a una persona en concreto.

Según McGraw-Hill (2015) nos da los siguientes conceptos sobre entidad-relación:

El modelo de datos entidad-interrelación (E-R), también llamado entidad-relación, fue propuesto por Peter Chen en 1976 para la representación conceptual de los problemas del mundo real. En 1988, el ANSI lo seleccionó como modelo estándar para los sistemas de diccionarios de recursos de información. Es un modelo muy extendido y potente para la representación de los datos. Se simboliza haciendo uso de grafos y de tablas. Propone el uso de tablas bidimensionales para la representación de los datos y sus relaciones. Los principales conceptos básicos son:

- **Entidad.** Es un objeto del mundo real, que tiene interés para la empresa. Por ejemplo, los ALUMNOS de un centro escolar o los CLIENTES de un banco. Se representa utilizando rectángulos.
- **Conjunto de entidades.** Es un grupo de entidades del mismo tipo, por ejemplo, el conjunto de entidades cliente. Los conjuntos de entidades no necesitan ser disjuntos, se puede definir los conjuntos de entidades de empleados y clientes de un banco, pudiendo existir una persona en ambas o ninguna de las dos cosas.

- **Entidad fuerte.** Es aquella que no depende de otra entidad para su existencia. Por ejemplo, la entidad ALUMNO es fuerte pues no depende de otra para existir, en cambio, la entidad NOTAS es una entidad débil pues necesita a la entidad ALUMNO para existir. Las entidades débiles se relacionan con la entidad fuerte con una relación uno a varios. Se representan con un rectángulo con un borde doble.

- **Atributos o campos.** Son las unidades de información que describen propiedades de las entidades. Por ejemplo, la entidad ALUMNO posee los atributos: número de matrícula, nombre, dirección, población y teléfono. Los atributos toman valores, por ejemplo, el atributo población puede ser ALCALÁ, GUADALAJARA, etcétera. Se representan mediante una elipse con el nombre en su interior.

- **Dominio.** Es el conjunto de valores permitido para cada atributo. Por ejemplo, el dominio del atributo nombre puede ser el conjunto de cadenas de texto de una longitud determinada.

- **Identificador o superclave.** Es el conjunto de atributos que identifican de forma única a cada entidad. Por ejemplo, la entidad EMPLEADO, con los atributos Número de la Seguridad Social, DNI, Nombre, Dirección, Fecha nacimiento y Tlf, podrían ser identificadores o superclaves los conjuntos Nombre, Dirección, Fecha nacimiento y Tlf, o también DNI, Nombre y Dirección, o también Num Seg Social, Nombre, Dirección y Tlf, o solos el DNI y el Número de la Seguridad Social.

- **Clave candidata.** Es cada una de las superclaves formadas por el mínimo número de campos posibles. En el ejemplo anterior, son el DNI y el Número de la Seguridad Social.

- **Clave primaria o principal (primary key):** Es la clave candidata seleccionada por el diseñador de la BD. Una clave candidata no puede contener valores nulos, ha de ser sencilla de crear y no ha de variar con el tiempo. El atributo o los atributos que forman esta clave se representan subrayados.

- **Clave ajena o foránea (foreign key):** Es el atributo o conjunto de atributos de una entidad que forman la clave primaria en otra entidad. Las claves ajenas van a representar las relaciones entre tablas. Por ejemplo, si tenemos por un lado, las entidades ARTÍCULOS, con los atributos código de artículo (clave primaria), denominación, stock. Y, por otro lado, VENTAS, con los atributos Código de venta (clave primaria), fecha de venta, código de artículo, unidades vendidas, el código de artículo es clave ajena pues está como clave primaria en la entidad ARTÍCULOS.

Base de datos No Relacional (NoSQL)

Según Gaimil Villarreal (2017) las bases de datos relacionales descansan su manejo en tablas, columnas y filas o esquemas para organizar y obtener los datos. Las bases de datos NoSQL no siguen este tipo de organización, sino que pueden acomodar los datos en diferentes modelos de datos tales como llave-valor, documentos, familia de columnas y formato de grafos. Estas bases de datos son particularmente útiles para almacenar y trabajar con grandes volúmenes de datos no estructurados, cuyo uso está aumentando a una velocidad mayor que los datos estructurados y que no tienen cabida en los esquemas relacionales. La mayoría son de código abierto.

Según Scofield (2009), una manera de clasificar los SGBD NoSQL es la forma en que almacenan los datos:

- **Clave/valor:** el almacenamiento se hace mediante estructuras con dos componentes: una clave y un valor. Ejemplo: {"nombre": "Juan"}, donde la clave es "nombre" y su valor es "Juan". Entre los SGBD NoSQL de este tipo están Redis, Dynamite! y Voldemort DB.
- **Columna:** el almacenamiento se hace por columnas. Es decir, una fila de datos corresponde a un grupo de valores para un mismo atributo (Figura 1). Por otro lado, en una BD relacional una fila de datos corresponde a un grupo de valores, uno para

cada atributo del esquema (Figura 2). Entre los SGBD NoSQL de este tipo están Apache Hbase, Cassandra y Hypertable.

Figura 1. Ejemplo de filas de empleados en una BD NoSQL orientada a columnas

1	2	3	→ Una fila de ids
Juan	Pablo	Ana	→ Una fila de Nombres
23	25	21	→ Una fila de Edades
8	10	6	→ Una fila de Grados
10	10	20	→ Una fila de Deps

Fuente: “Rendimiento entre Oracle y MongoDB”. <http://dx.doi.org/10.18359>

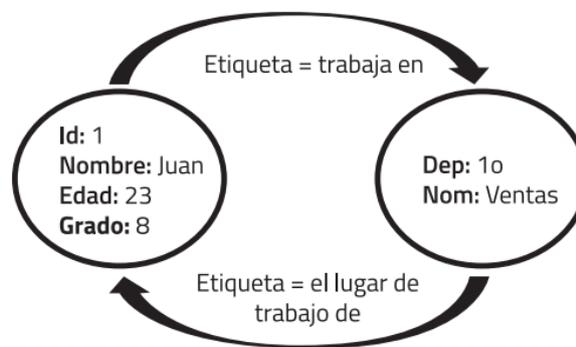
Figura 2. Ejemplo de filas empleados en un BD relacional orientada a filas

ID	NOMBRE	EDAD	GRADO	DEP	
1	Juan	23	8	10	→ Un empleado
2	Pablo	25	10	10	→ Un empleado
3	Ana	21	6	20	→ Un empleado

Fuente: “Rendimiento entre Oracle y MongoDB”. <http://dx.doi.org/10.18359>

- **Grafo:** el almacenamiento se hace mediante grafos. En los nodos se almacenan entidades, las cuales se relacionan por medio de aristas. Ejemplo: Ver Figura 3. Entre los SGBD NoSQL de este tipo están Neo4j, OrientDB e InfiniteGraph.

Figura 3. Representación de datos en una BD NoSQL de tipo grafo



Fuente: “Rendimiento entre Oracle y MongoDB”. <http://dx.doi.org/10.18359>

- **Documento:** el almacenamiento se hace en colecciones de documentos, es decir, los documentos JSON (JavaScript Object Notation). Entre los SGBD NoSQL de este tipo están MongoDB y Apache CouchDB.

COMPARATIVA ENTRE SQL Y NOSQL

A continuación, en la Tabla 1 se compara las características de las bases de datos relacionales y no relacionales y la forma de uso entre los SGBD MongoDB y Oracle. Se eligió ambas bases de datos por ser uno de los principales SGBD en el mercado (véase la clasificación y ranking en <http://db-engines.com/en>).

Tabla 1. Principales características y diferencias entre las bases de datos SQL y NOSQL

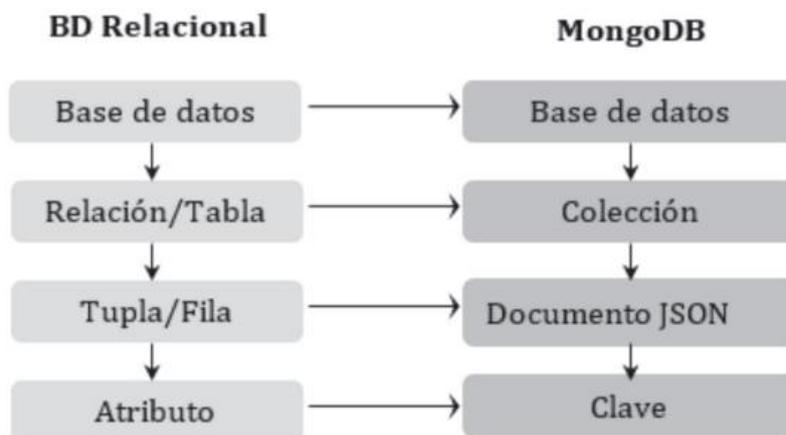
CARACTERÍSTICAS	BASE DE DATOS RELACIONALES SQL	BASE DE DATOS NOSQL
Modelo de datos	El modelo relacional almacena los datos siguiendo las 3 formas de normalización, en estructuras de tablas, las cuales están formadas por filas y columnas. Trabajan con un esquema que define estrictamente en objetos las tablas, columnas, índices,	Las bases de datos no relacionales (NoSQL) no contienen un esquema tan rígido como el de las bases de datos relacionales. Mayormente se utiliza una llave de partición para recuperar valores, conjuntos de columnas o
CARACTERÍSTICAS	BASE DE DATOS RELACIONALES SQL	BASE DE DATOS NOSQL

Modelo de datos	relaciones entre las tablas y otros elementos que forman parte de las bases de datos.	documentos JSON o XML semiestructurados,
Propiedades ACID (Teorema CAP)	Los sistemas de administración de bases de datos relacionales (RDBMS) tradicionales manejan un conjunto de propiedades definidas por el acrónimo ACID (por sus siglas en inglés) que significan Atomicidad (Atomicity), Consistencia (Consistency), Aislamiento (Isolation) y Durabilidad (Durability).	Normalmente, las bases de datos NoSQL intercambian algunas de las propiedades ACID de los sistemas de RDBMS tradicionales por un modelo de datos más flexible que escala en forma horizontal.
Desempeño	Para lograr un desempeño acorde con las necesidades, es necesaria la optimización de consultas, índices y los valores de almacenamiento de las estructuras de tablas, así como la correcta disposición de los datos físicos en los discos.	El desempeño, por lo general, tiene dependencia con el tamaño del cluster de hardware utilizado, la latencia de red y la aplicación que efectúa la llamada.
Escalamiento	Para lograr escalabilidad se necesita manejar disponibilidad de recursos físicos en hardware más rápido y manejo de estructuras en memoria.	Está diseñada para reducir el escalamiento usando clusters distribuidos de hardware de bajo costo para aumentar el desempeño sin que aumente la latencia.
CARACTERÍSTICAS	BASE DE DATOS RELACIONALES SQL	BASE DE DATOS NOSQL
Escalamiento	Se requieren inversiones adicionales para que las tablas relacionales abarquen un sistema distribuido.	
API	Solicita almacenar y recuperar datos que están	Las API basadas en objetos permiten a los

	comunicados mediante consultas que se ajustan a un lenguaje de consulta estructurado (SQL por sus siglas en inglés). Estas consultas son analizadas y ejecutadas por los sistemas RDBMS.	desarrolladores almacenar y recuperar fácilmente estructuras de datos en memoria. Las llaves de partición permiten que las aplicaciones busquen pares de llave-valor, familias de columnas o documentos semiestructurados que contengan atributos y objetos de aplicación serializados.
Disponibilidad de Herramientas	Las bases de datos SQL normalmente ofrecen un amplio conjunto de herramientas que simplifican el desarrollo de aplicaciones de base de datos.	Las bases de datos NoSQL suelen ofrecer herramientas para administrar los clusters y el escalamiento. Las aplicaciones representan la interfaz primaria de los datos subyacentes.

Fuente: <https://aws.amazon.com/es/nosql/>

Figura 4. Comparación de términos entre una BD relacional y una BD orientada a documentos (MongoDB)



Fuente: “Rendimiento entre Oracle y MongoDB”. <http://dx.doi.org/10.18359>

En la Figura 4 se puede observar las equivalencias de términos que se utilizan para definir los conceptos básicos de una base de datos relacional y no relacional. Por ejemplo, en las bases de datos no relacional las tablas se conocen como Colecciones, y un Atributo se conoce como Clave.

Figura 5. Formas generales de las sentencias básicas de definición del esquema en SQL y en MongoDB

Operación	SQL	MongoDB
Creación base de datos	CREATE DATABASE nombreBD;	USE nombreBD
Creación tabla (en SQL) / Colección (en MongoDB)	CREATE TABLE nombreTabla (atributo1 tipo_de_dato restricciones <, atributo2...>);	db.createCollection ("nombreColección")
Creación índice	CREATE INDEX nombreIndice ON nombreTabla (atributo1 <, atributo2... >);	db.nombreColeccion.ensureIndex ({"atributo": 1})
Dstrucción tabla (en SQL) / Colección (en MongoDB)	DROP TABLE nombreTabla;	db.nombreColeccion.drop ()

Fuente: “Rendimiento entre Oracle y MongoDB”. <http://dx.doi.org/10.18359>

En la Figura 5 se muestra la diferencia de definiciones para la creación de una base de datos, creación de tablas o colecciones, creación de índices y destrucción de tablas o colecciones de las bases de datos SQL y NoSQL.

Figura 6. Ejemplos de las sentencias de definición del esquema en SQL y en MongoDB

SQL	MongoDB
CREATE DATABASE empresa;	USE empresa
CREATE TABLE dpto (dep NUMBER(3) PRIMARY KEY , nom VARCHAR(20) NOT NULL); CREATE TABLE empleado (id NUMBER(4) PRIMARY KEY , nombre VARCHAR(20) NOT NULL , edad VARCHAR(20) NOT NULL , grado NUMBER(3) NOT NULL , dep NUMBER(3) REFERENCES dpto);	db.createCollection("dpto") db.createCollection("empleado")
CREATE INDEX indiceDep ON empleado (dep);	db.empleado.ensureIndex({dep: 1}) //Suponiendo que empleado tiene una clave dep.
DROP TABLE empleado;	db.empleado.drop()

Fuente: "Rendimiento entre Oracle y MongoDB". <http://dx.doi.org/10.18359>

En la Figura 6 se puede observar con un ejemplo la diferencia entre las sentencias de SQL y MongoDB para la creación de tablas, colecciones, la forma como se generan índices y el borrado de estas respectivamente.

Figura 7. Formas generales de las sentencias de inserción, actualización y borrado de datos en SQL y en MongoDB

Operación	SQL	MongoDB
Inserción	INSERT INTO nombreTabla VALUES (valorAtributo1, <, valorAtributo2...>);	db.nombreColeccion.insert({ atributo1: valorAtributo1 <, atributo2: valorAtributo2, ...>})
Actualización	UPDATE nombreTabla SET atributo = nuevoValor < WHERE condición>	db.nombreColeccion.update({<condición>}, {\$set: {atributo: nuevoValor} }<, [true false],[true false]>)
Borrado	DELETE FROM nombreTabla < WHERE condición>	db.nombreColeccion.remove({<condición>})

Fuente: "Rendimiento entre Oracle y MongoDB". <http://dx.doi.org/10.18359>

En la Figura 7 se muestra las sintaxis en SQL y MongoDB para la inserción, actualización y borrado de tablas y colecciones respectivamente.

Figura 8. Ejemplos de las sentencias de inserción, actualización y borrado de datos en SQL y en MongoDB

SQL	MongoDB	Resultados en MongoDB
INSERT INTO dpto VALUES (10, 'Ventas');	db.dpto.insert({_id: 10, nom: "Ventas"})	Los datos son insertados.
INSERT INTO empleado VALUES (1, 'Juan', 23, 8, 10);	db.empleado.insert({_id: 1, nombre: "Juan", edad: 23, grado: 8, dep: 10})	
UPDATE empleado SET grado = 9 WHERE edad = 25;	db.empleado.update({edad:25}, {\$set: {grado: 9}})	{ "_id": 1, "nombre": "Juan", "edad": 23, "grado": 8, "dep": 10} { "_id": 2, "nombre": "Pablo", "edad": 25, "grado": 9, "dep": 10} { "_id": 3, "nombre": "Ana", "edad": 11, "grado": 6, "dep": 20}
DELETE FROM empleado WHERE edad >= 18;	db.empleado.remove({edad: {\$gt:18}})	{ "_id": 3, "nombre": "Ana", "edad": 11, "grado": 6, "dep": 20}

Fuente: "Rendimiento entre Oracle y MongoDB". <http://dx.doi.org/10.18359>

En la Figura 8 se muestra un ejemplo donde se realiza la inserción, actualización y borrado de datos en SQL y en MongoDB, además en la tercera columna se puede observar el resultado obtenido en MongoDB.

Figura 9. Ejemplos de las consultas en SQL y en MongoDB

SQL	MongoDB	Resultado en MongoDB
SELECT * FROM empleado WHERE edad < 30 AND grado = 8;	db.empleado.find({edad: {\$lt: 30}, grado: 8})	{ "_id": 1, "nombre": "Juan", "edad": 23, "grado": 8, "dep": 10}
SELECT * FROM empleado WHERE nombre IN ('Pablo', 'Ana');	db.empleado.find({nombre: {\$in: ["Pablo", "Ana"]}})	{ "_id": 2, "nombre": "Pablo", "edad": 25, "grado": 10, "dep": 10} { "_id": 3, "nombre": "Ana", "edad": 11, "grado": 6, "dep": 20}
SELECT * FROM empleado WHERE nombre LIKE '%blo';	db.empleado.find({nombre: /blo/})	{ "_id": 2, "nombre": "Pablo", "edad": 25, "grado": 10, "dep": 10}
SELECT SUM (edad) AS total FROM empleado;	db.empleado.aggregate([{\$group: { _id: null, total: {\$sum: "\$edad"} } }])	[{ "_id": null, "total": 69 }]
SELECT dep, MAX (grado) AS mg FROM empleado GROUP BY dep;	db.empleado.aggregate({\$group: { _id: "\$dep", mg: {\$max: '\$grado'} } })	[{ "_id": 10, "max": 10 }, { "_id": 20, "mg": 6 }]

Fuente: "Rendimiento entre Oracle y MongoDB". <http://dx.doi.org/10.18359>

En la Figura 9 se muestra un ejemplo de consultas en SQL y MongoDB, se puede observar que para manipulación las de las bases de datos relacionales es necesario conocer las sentencias SQL, mientras que, para manipular MongoDB es necesario conocer JSON. Otras bases de datos NoSQL utilizan Javascript para la manipulación de los datos.

DESCRIPCIÓN DEL PROTOCOLO DE INVESTIGACIÓN

El protocolo de investigación especificará los métodos que se utilizarán para llevar a cabo la estrategia de búsqueda, estos son: los criterios de selección de documentos, evaluación de calidad y el proceso de extracción o recopilación.

El tipo de estudio de esta revisión es analítico y de comparación. Toma como referencia la revisión y recopilación sistemática de artículos de interés para luego identificar cuáles serán relevantes para el estudio.

En la tabla 2 se muestra un resumen sobre la estrategia de búsqueda aplicada a la presente revisión sistemática. Primero, se establece la idea de estudio (bases de datos en las empresas), segundo, se utiliza la metodología PICO (población, intervención, control y resultado) que nos da como resultado las palabras base de búsqueda, tercero, se formula la pregunta de investigación, cuarto, obtenemos las variables de búsqueda, y en los siguientes puntos se forman las cadenas para realizar las búsquedas, por último, se indica los repositorios donde se van a realizar las búsquedas de información.

Tabla 2. Resumen de la Estrategia de Búsqueda

Idea de estudio	Bases de Datos Empresarial	
Problema de estudio PICO	Población Intervención Control Resultado	Empresas NOSQL SQL Diferencias entre SQL y NOSQL
Pregunta	¿Cuáles son las diferencias entre una base de datos SQL y NOSQL en el ámbito empresarial?	
VARIABLES DE BÚSQUEDA O TÉRMINOS BÁSICOS	Empresas NOSQL SQL Bases de datos	
Planteamiento de búsquedas por variables	Empresas, Negocios, Compañías NOSQL, No relacional SQL, Relacional Bases de datos	(Empresas OR Negocios OR Compañías) (NOSQL OR No relacional) (SQL OR Relacional) (Bases de datos)
Búsqueda avanzada	(Empresas OR Negocios OR Compañías) AND (NOSQL OR No relacional) AND (SQL OR Relacional) AND (Bases de datos)	

Fuente: Elaborado a partir de Metodología de Investigación. R. Sampieri.

PROCESO DE BÚSQUEDA DE DOCUMENTOS

El proceso de búsqueda de información se realizó en dos fases: en la primera se optó por buscar información con términos básicos (Tabla 2) implicados con la búsqueda. En la segunda, con información refinada obtenida en la primera fase, se decidió por localizar información más específica, para mejorar las conclusiones y hacer un mejor estudio secundario. A continuación se desarrolla el proceso de búsqueda:

Términos básicos de búsqueda:

- Empresas, Negocios, Compañías
- NOSQL, No relacional
- SQL, Relacional
- Bases de datos

Luego los términos básicos similares se agruparon asignando el operador lógico OR.

- (Empresas **OR** Negocios **OR** Compañías)
- (NOSQL **OR** No relacional)
- (SQL **OR** Relacional)
- (Bases de datos)

Por último, se creó la cadena de búsqueda asignando a cada grupo de términos similares el operador lógico AND.

*(Empresas **OR** Negocios **OR** Compañías) **AND** (NOSQL **OR** No relacional) **AND**
(SQL **OR** Relacional) **AND** (Bases de datos)*

La lista de fuentes o repositorios de investigación científica sobre las cuales se realizaron las búsquedas son:

- Scielo
- Redalyc
- Redib

En la primera etapa, la búsqueda se realizó usando los términos básicos mencionados. Es aquí donde notamos que algunos de los términos usados también se encontraban en otras ramas de la ingeniería.

Tabla 3. Cantidad de artículos encontrados en primera fase de búsqueda

Keywords	Scielo	Redalyc	Redib
Bases de datos	7730	528758	10023
SQL	75	745	215
NoSQL	3	28	27
Total	7,808	529,531	10,265

Fuente: Propia

De lo mostrado en la Tabla 3 podemos mencionar que en la primera fase de búsqueda se encontraron 547,604 artículos que mencionan los términos de búsqueda.

PROCESO DE SELECCIÓN DE DOCUMENTOS

Para realizar el proceso de selección de búsqueda se han tomado los siguientes criterios de inclusión y exclusión:

Criterios de Selección Inclusión

- Fecha de antigüedad: 8 años (publicaciones posteriores o igual al año 2010)
- Tipo de publicación: Artículos y revistas.
- Idioma: español
- Lugar: América Latina y España

Criterios de Selección Exclusión

- Tipo de fuente: Tesis, trabajos de grado, ensayos.
- Idioma: Diferente al español e inglés

Tabla 4. Cantidad de artículos encontrados aplicando filtros de selección

Términos	Scielo	Redalyc	Redib
Bases de datos	3	14	26
SQL	6	116	73
NoSQL	2	16	12
Total	11	146	111

Fuente: Propia

En la Tabla 4 podemos observar que a pesar de haber aplicado los filtros de selección sigue obteniendo el mayor puntaje el repositorio de Redalyc con 146 seguido de Redib con 11 y por último Scielo 111.

Luego de realizar búsquedas con los filtros de inclusión y exclusión se han encontrado 258 artículos de los cuales solo 76 estaban relacionados al tema y 19 eran

artículos repetidos. En total quedaron 57 artículos a revisar para responder la pregunta de investigación (Ver Tabla 5).

Tabla 5. Resultados de artículos encontrados para la revisión

Filtro	Criterio de Selección	Relación al Tema	Repetidos	Revisión
	258	76	19	57

Fuente: Propia

PROCESO DE EVALUACIÓN DE CALIDAD

De los artículos encontrados, no todos cuentan con un indicador de evaluación de calidad propio del documento, los artículos que si traen una calificación de calidad no utilizan el mismo estándar de evaluación, algunos artículos utilizan los cuartiles y otros por ejemplo el número de visitas o cantidad de citas textuales.

Al no encontrar un patrón que estandarice o califique la calidad de los artículos se ha propuesto establecer (nosotros mismos) un criterio de ranking A, B y C. Donde la calificación A denota positivamente al artículo y respondería la pregunta de investigación en su totalidad, la calificación B denota una respuesta positiva intermedia pero que no satisface al 100% a la pregunta de investigación y la calificación C indica que el artículo responde breve y sutilmente a la pregunta.

PROCESO DE EXTRACCIÓN DE DATOS

En esta etapa se realiza el diseño de los formularios de extracción de datos para registrar la información obtenida de las bases de datos de investigación.

Los formularios de extracción de datos serán diseñados para recoger toda la información necesaria para poder responder a las preguntas de la revisión.

En total se recopilieron 57 artículos y revistas para su revisión.

A continuación, en la Tabla 6, presentamos un formato de tabla a usar para la presentación de la extracción de datos:

Tabla 6. Formato de tabla de extracción de datos

Nro. 1	
Palabras Claves	SQL, NOSQL
Fuente Búsqueda	Redib
Título	Sistemas para almacenar grandes volúmenes de datos
País	Colombia

Año	2014
Autor	Valbuena, S. J., & Londoño, J. M.
Tipo Publicación	Artículo
Referencia Bibliográfica	Valbuena, S. J., & Londoño, J. M. (2014). SISTEMAS PARA ALMACENAR GRANDES VOLÚMENES DE DATOS. Gerencia Tecnológica Informática, 13(37).
Calidad	A
URL	https://www.redib.org/recursos/Record/oai_articulo609941-sistemas-almacenar-vol%C3%BAmenes-datos

Fuente: Propia

A partir de la información extraída de los estudios realizó un análisis para poder mostrar los datos estadísticos más importantes de la revisión sistemática hecha.

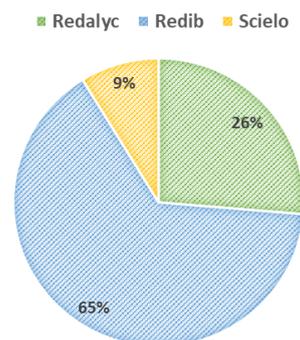
La Tabla 7 y Figura 10 muestra la cantidad de artículos y revistas por base de datos, se puede observar que Redib es el repositorio donde más información se obtuvo.

Tabla 7. Distribución de artículos por fuente

Base de Datos	Total	%
Redalyc	15	26%
Redib	37	65%
Scielo	5	9%
Total general	57	100%

Fuente: Propia

Figura 10. Gráfico de artículos por fuente



Fuente: Propia

La Tabla 8 muestra la distribución anterior segmentados por artículos (papers científicos) y revistas. Se puede observar que la mayor información se ha extraído de revistas (47).

Tabla 8. Distribución de base datos por artículo y revistas

Etiquetas de fila	Artículo	Revista	Total
Redalyc	4	11	15
Redib	3	34	37
Scielo	3	2	5
Total general	10	47	57

Fuente: Propia

La Tabla 9 muestra la distribución del término de búsqueda (palabra clave) por repositorio o fuente. La palabra clave que más ha dado resultado es “Base de datos”.

Tabla 9. Distribución de termino clave por base de datos

Termino Clave	Redalyc	Redib	Scielo	Total
Bases de datos	6	26	2	34
NoSQL	5	7		12
SQL	4	4	3	11
Total general	15	37	5	57

Fuente: Propia

La Tabla 10 muestra cantidad papers encontrados por cada país y año (2010 – 2018). Colombia y Cuba han producido más documentación sobre el tema de bases de datos SQL y NOSQL a la actualidad. La producción investigadora de Perú sobre el tema es baja al igual que Venezuela en comparación a los demás países de habla hispana, incluido Brasil.

Tabla 10. Distribución de artículos encontrados por países

Etiquetas de fila	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total	%
Brasil				1				1		2	4%
Colombia	5	2	2	1	3	3	3	3		22	39%
Costa rica			1							1	2%
Cuba	1	3	5	3	2		5		1	20	35%
Ecuador							1		1	2	4%
España	1									1	2%
México				2	1	1		1		5	9%
Panamá			1			1				2	4%
Perú								1		1	2%
Venezuela				1						1	2%
Total general	7	5	9	8	6	5	10	5	2	57	100%

Fuente: Propia

La Tabla 11. muestra los artículos incluidos y excluidos según base de datos. En total son 26 artículos incluida que justifican la producción esta revisión sistemática.

Tabla 11. Distribución de artículos incluido y excluidos según base de datos

Filtro	Redalyc	Redib	Scielo	Total	%
Excluye	6	17	3	26	46%
Incluye	9	20	2	31	54%
Total general	15	37	5	57	100%

Fuente: Propia

Por último, la Tabla 12 muestra un índice o ranking de calidad de los artículos incluidos para la justificación de estudio realizado. Cabe indicar que en el cuadro anterior (Tabla. 11) los artículos excluidos han sido seleccionados porque no respondían la pregunta del tema.

En la Tabla 12, las letras A, B y C indican el nivel de contenido satisfactorio que responden a la pregunta de investigación, donde A es el nivel mas alto (nivel de respuesta satisfactorio) y C el nivel más bajo.

Tabla 12. Distribución de artículos incluido y excluidos según base de datos

Ranking Calidad	Redalyc	Redib	Scielo	Total Calidad	%
A	5	10	1	16	52%
B	2	6	1	9	29%
C	2	4	0	6	19%
Total general	9	20	2	31	100%

Fuente: Propia

CAPÍTULO III. RESULTADOS

A continuación, se describen brevemente y se recogen los datos más destacables.

En el artículo de Moreno, Quintero y Rueda (2016) se muestra una comparación de rendimiento entre Oracle y MongoDB. Aunque se requieren experimentos más exhaustivos y otros tipos de pruebas más rigurosas, los tiempos registrados para las operaciones de inserción, actualización, borrado y consultas sobre una sola tabla (SQL) / colección (NoSQL) favorecieron a MongoDB. Esto se debe posiblemente al mayor número de verificaciones de consistencia, de integridad y a la gestión de transacciones realizadas por Oracle.

Sin embargo, para operaciones de consulta binarias, es decir, que involucran dos relaciones / colecciones como la reunión (join), el tiempo de respuesta favoreció a Oracle. Esto se debe posiblemente a los costos implicados para la ejecución de la reunión por medio de la programación adicional requerida.

El estudio de Jaramillo (2014) podemos evidenciar que a pesar de la madurez que han alcanzado los sistemas de gestión de base de datos relacional en los últimos años, hay aspectos relativos en cuanto a su desempeño que aún no han sido resueltos, uno de ellos es el escalamiento (horizontal y vertical). Esta limitación es un punto de inflexión que ha motivado el surgimiento de otro tipo de sistemas como el NoSQL, los cuales pueden escalar horizontalmente, debido a su flexibilidad, al particionamiento de datos (sharding), al buen manejo del concepto de redundancia y a la capacidad de procesar tareas concurrentemente.

CAPÍTULO IV. CONCLUSIONES

En conclusión, la diferencia fundamental entre SQL y NoSQL no es tan complicada, cada una tiene una filosofía diferente sobre cómo se deben almacenar y recuperar los datos. Por lo tanto, para poder diferenciar correctamente una base de datos SQL y NOSQL dependerá del proyecto de desarrollar. Muchas personas piensan que las tecnologías NoSQL son lo “nuevo” y por lo tanto todo debe migrar a este modelo, pero es un grave error. NoSQL no es un reemplazo, es simplemente un modelo diferente que ofrece ventajas y soluciones a problemas que poseen las bases de datos relacionales.

En el proyecto que necesite respetar la integridad de los datos, por ejemplo, una aplicación para un banco, inteligencia y análisis de negocios para la identificación de patrones en los datos, software a medida, desarrollos web o software empresarial que necesitan información con estructura consistente o de necesitar un mayor soporte, entonces el proyecto debe optar por una base de datos SQL.

Sin embargo, los proyectos que necesiten una gran cantidad de escalabilidad y donde los recursos son escasos (hardware, procesamiento, etc.) y no necesitan respetar la integridad de los datos, por ejemplo, una aplicación para internet de las cosas (IOT) con millones de dispositivos reportando los datos de sensores, micrófonos, cámaras, escáneres médicos, imágenes, etc. o redes sociales, además al no contar con un tipo de dato específico, es más abierta y flexible, entonces el proyecto debe optar por una base de datos NoSQL.

REFERENCIAS

- [1]Antaño, A. C. M., Castro, J. M. M., & Valencia, R. E. C. (2014). Migración de Bases de Datos SQL a NoSQL. *Revista Tlamati, Especial*, 3, 144-148.
- [2]Rivera, F. L. O. (2008). Base de datos relacionales. ITM.
- [3]Herrera, H. A., & Valenzuela, C. R. (2016). NoSQL, la nueva tendencia en el manejo de datos. *Tecnología Investigación y Academia*, 4(1), 147-150.
- [4]Pollo, M., López, M. & Daián, G. (2014) Rendimiento de tecnologías NoSQL sobre cantidades masivas de datos. *Cuaderno Activa*, 6, pp11-17.
- [5]Bas Abad, VJ. (2015). Estudio comparativo de BBDD relacionales y NoSQL en un entorno industrial.
- [6]Morata López, Borja (2018). Análisis comparativo entre un sistema de Business Intelligence sobre una base de datos SQL y NoSQL. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S.I. de Sistemas Informáticos (UPM), Madrid.
- [7]Minango Román, Carlos Edmundo y Tenelema Córdor, Cristian Jefferson (2018). Análisis del uso de recursos entre bases de datos columnares versus bases de datos relacionales. Trabajo de titulación previo a la obtención del Título de Ingeniero Informático. Carrera de Ingeniería Informática. Quito: UCE. 65 p.
- [8]Tello Y., R. (2003). Base de datos en la ingeniería y los negocios. *Industrial Data*, 6 (1), 79-82.
- [9]Castro Romero, A., & González Sanabria, J., & Callejas Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21 (33), 21-32.
- [10]Castro Romero, A., & González Sanabria, J., & Callejas Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21 (33), 21-32.

- [11]Castro Romero, Alexander, González Sanabria, Juan Sebastián, Callejas Cuervo, Mauro, Utilidad y funcionamiento de las bases de datos NoSQL. Facultad de Ingeniería [en línea] 2012, 21 (Julio-Diciembre) : [Fecha de consulta: 7 de octubre de 2018] Disponible en:<<http://www.redalyc.org/articulo.oa?id=413940772003>> ISSN 0121-1129
- [12]Hernández-Leal, E., & Duque-Méndez, N., & Moreno-Cadavid, J. (2017). Big Data: una exploración de investigaciones, tecnologías y casos de aplicación. *TecnoLógicas*, 20 (39)
- [13]Pérez, M. A. C. (2017). NoSQL, ¿es necesario ahora? *Tecnología, Investigación y Academia*, 5(2).
- [14]Álvarez, C. C. P., Pinilla, C., & Bello, M. (2017). Bases de datos orientadas a grafos. *Tecnología, Investigación y Academia*, 5(2).
- [15]Valbuena, S. J., & Londoño, J. M. (2014). Sistemas para almacenar grandes volúmenes de datos. *Gerencia Tecnológica Informática*, 13(37).
- [16]Arboleda, F. J. M., Rendón, J. E. Q., & Vásquez, R. R. (2016). Una comparación de rendimiento entre Oracle y MongoDB. *Ciencia e Ingeniería Neogranadina*, 26(1)
- [17]F. J. Moreno Arboleda, J. E. Quintero Rendón, R. Rueda Vásquez. (2016). Una comparación de rendimiento entre Oracle y MongoDB. *Ciencia e Ingeniería Neogranadina*, 26 (1), pp. 109-129, DOI: <http://dx.doi.org/10.18359/rcin.1669>.
- [18]Sadalage, P. & Fowler, M. (2013) *NoSQL Distilled, A Brief Guide to the Emerging World of Polyglote*. Persistence Boston: Addison Wesley.
- [19]Pollo, M., López, M. & Daián, G. (2014) Rendimiento de tecnologías NoSQL sobre cantidades masivas de datos. *Cuaderno Activa*, 6, pp11-17.
- [20]Jaramillo Valbuena, S. & Londoño, J. M. (2014). Sistemas para almacenar grandes volúmenes de datos. En R, Llamasa Villalba (Ed.). *Revista Gerencia Tecnológica Informática*, 13(37), 17-28. ISSN 1657-8236.