## Resumen

A partir del nacimiento del internet durante los años 80s, las bases de datos SQL dominaron el mercado, cualquier tipo de aplicación, tecnología, etc. Se desarrollaba con SQL, sin embargo, a lo largo de los años el internet evolucionó, la cantidad de datos, la escala y la forma de hacer las cosas cambiaron, esto hizo que empresas, organizaciones, y muchas personas adoptaran las bases de datos NoSQL como la tecnología adecuada para las nuevas necesidades. Estas bases de datos están diseñadas para ser usadas en grandes sistemas distribuidos, son notablemente mucho más escalables además de ser más rápidas que las bases de datos tradicionales manejando cargas de datos muy grandes. Este tipo de base de datos almacena datos en un formato distinto a las clásicas tablas relacionales, se utilizan de forma generalizada en aplicaciones web en tiempo real y big data. En esta investigación se comprobará cuál base de datos no relacional es la que tiene mejor rendimiento de acuerdo a los parámetros y condiciones de la investigación.


## I. Introducción

Desarrollar y utilizar las nuevas aplicaciones de la actualidad ha creado nuevas necesidades en la arquitectura de las bases de datos NoSQL, estas tienen que ser cada vez más ágiles, también requieren un desarrollo cada vez más enfocado a los datos en tiempo real, al igual que cada vez es más necesario que esta tecnología pueda procesar cómodamente impredecibles niveles de escala, velocidad y variabilidad de datos, agregando a todo esto la necesidad de las empresas y organizaciones de innovar rápidamente, operar a cualquier escala, además de cumplir la demanda principal que es la experiencia de usuario.

Estas bases de datos ofrecen diferentes tipos de modelos de datos para almacenar la información que son ideales para construir aplicaciones que requieren el manejo de grandes cantidades de información con una latencia baja de respuesta, ya que sus principales ventajas son los elevados niveles de escalabilidad y disponibilidad, además de ser ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento. Por lo cual es de suma importancia conocer las opciones al igual que las alternativas que existan en el mercado, también ser consciente de las fortalezas y necesidades de cada una de ellas. Al ser una tecnología relativamente nueva, cada día estas herramientas naturalmente se actualizan, por lo cual estar a la vanguardia ayuda a siempre elegir la mejor opción de acuerdo a las necesidades requeridas, esta investigación se enfoca en las bases de datos NoSQL que almacenan los datos mediante a documentos.

## II. Marco Teórico

*2.1* Base de datos NoSQL.

Las bases de datos NoSQL son una categoría de Sistemas de Gestión de Bases de Datos que no utilizan SQL como lenguaje de consulta principal. Estas bases de datos no requieren esquemas de tablas fijas y no soportan operaciones Join. Están optimizadas para operaciones de lectura/escrituras escalables en lugar de pura consistencia. Asimismo, constituyen un ecosistema de información y se están convirtiendo en alternativas viables a las bases de datos relacionales para muchas aplicaciones. En el apartado siguiente, dedicado al estudio en profundidad de esta categoría de bases de datos. (Aguilar, 2019)

La repuesta a la necesidad de gestionar volúmenes masivos de información surge de la base de datos NoSQL, termino acuñado a finales de los 90 y que engloba todas las tecnologías de almacenamiento estructurado que no cumplen el esquema relacional. La cantidad de información manejada por comunidades, redes sociales, buscadores y muchos otros proyectos en el ámbito de la Web 2.0 es abrumadora, lo que ha hecho que surjan nuevas arquitecturas de almacenamiento de información, que deben ser de alto rendimiento, escalables y distribuidas. Aunque esta tecnología surgió de unas necesidades muy concretas, su difusión y algunos proyectos para encapsular sus funcionalidades y hacerlas más amigables a los desarrolladores acostumbrados a SQL está provocando que también se usen en proyectos de pequeño tamaño, con lo que todo indica que a medio plazo convivirán con las bases de datos tradicionales independientemente del volumen de datos a gestionar. (Gracia del Busto & Yanes Enríquez, 2012)

Desde su creación, las bases de datos han sido un soporte para la organización de la información dentro de los diferentes tipos de entidades, debido a que “las bases de datos comenzaron a aparecer a finales de 1950 y comienzos de 1960, impulsadas por dos factores tecnológicos: el incremento de la fiabilidad de los procesadores de ordenador y la expansión de la capacidad de almacenamiento secundario en cintas y unidades de disco” (J. BERNDT, LASA, & MCCART, 2012)

## *Concepto NoSQL*

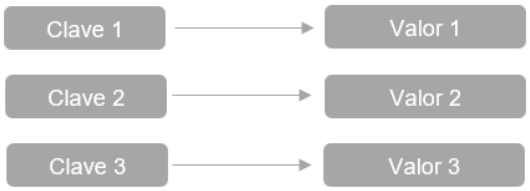
Desde la aparición del término NoSQL existe un inconveniente conceptual (en un principio se pensó usar el término “NoRel”, haciendo referencia a “No Relacional”, pero no sonaba comercial), puesto que la denominación puede interpretarse a primera vista como oposición de SQL; por ello se ha querido dar vuelta a este concepto aclarando que NoSQL (siglas en inglés de Not only SQL) se define como “No solo SQL”; (C. Romero, G. Sanabria, & C. Cuervo, 2012) “NoSQL es usado como un término general por todas las bases de datos y almacenes de datos que no siguen los populares y bien establecidos principios RDBMS (Relational Database Management System), y a menudo está relacionado con grandes conjuntos de datos y su manipulación en una escala Web” (Tiwari, 2011 )

## *Patrones de arquitectura de datos NoSQL.*

**Clave Valor**: Los sistemas de clave-valor prometen un rendimiento excelente para volúmenes de datos muy grandes, a cambio de ser muy simples y renunciar a

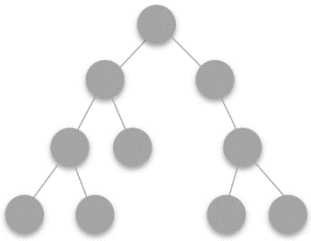
funcionalidades que tenemos en otros sistemas como la verificación intrínseca de la integridad de datos, llaves extranjeras y disparadores. Las validaciones de los datos se delegan completamente en la aplicación cliente, siendo la base de datos, simplemente el lugar donde se guardan los datos. No se verifican integridades, no se comprueban referencias cruzadas, todo esto se ha de implementar a nivel de aplicación, en el código del cliente. (Gracia del Busto & Yanes Enríquez, 2012)

*Figura 1. Diagrama representa el tipo de dato Clave\Valor*
*En bases de datos NoSQL*




**Documentos**: Una base de datos orientada a documentos está diseñada para gestionar información orientada a documentos o datos semi-estructurados. Este tipo de bases de datos constituye una de las principales categorías de las llamadas bases de datos NoSQL. La popularidad del término "base de datos orientada a documentos" o "almacén de documentos" ha crecido a la par con el uso del término NoSQL en sí. A diferencia de las conocidas bases de datos relacionales con su definición de "tabla", los sistemas documentales están diseñados entorno a la definición abstracta de un "documento". (Gracia del Busto & Yanes Enríquez, 2012)

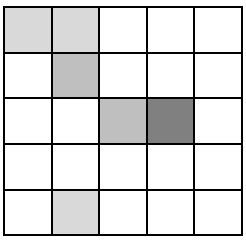
*Figura 2. Diagrama representa el tipo de dato Documentos*
*En bases de datos NoSQL*



**Columnas Anchas**: Las bases de datos orientadas a columnas son probablemente más conocidas por la aplicación BigTable de Google o por la implementación Cassandra de Apache. A primera vista son muy similares a las bases de datos relacionales, pero en realidad son muy diferentes. Una de las principales diferencias radica en el almacenamiento de datos por filas (sistema relacional) versus el almacenamiento de datos por columnas (sistema orientado a columnas) y otra la optimización de consultas para mejorar los tiempos de respuesta en comparación con los sistemas relacionales. (Gracia del Busto & Yanes Enríquez, 2012)

*Figura 3. Diagrama representa el tipo de dato Columnas Anchas*

*En bases de datos NoSQL*



**Grafos**: Las bases de datos orientadas a grafos representan la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones). (Gracia del Busto & Yanes Enríquez, 2012)

*Figura 4. Diagrama representa el tipo de dato Grafos*
*En bases de datos NoSQL*

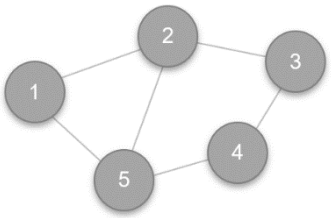



*Tabla 1. Bases de datos NoSQL más populares según su tipo de dato*

| Clave \ Valor | Documentos | Columnas Anchas | Grafos |
|---|---|---|---|
| Redis | MongoDB | Hbase | Neo4j |
| Amazon DynamoDB | Databricks | Cassandra | Virtuoso |
| Hazelcast | Amazon DynamoDB | | ArangoDB |
| Memcached | Microsoft Azure CosmosDB | | OrientDB |

## *2.2 Benchmarking*

Según (Barzu, 2017), El concepto de benchmark no es nada nuevo en la informática. En los años 1970, el concepto de benchmark se formó como un término técnico que significa "punto de referencia". Posteriormente este término migró al ámbito empresarial, donde se definió el benchmark como el proceso de medir para realizar comparaciones.

Al proceso de comparar dos o más sistemas mediante la obtención de medidas se denomina benchmarking. Podemos considerarlo como un proceso cuyo fin es la optimización de los resultados a través del estudio, adaptación e implantación de métodos ya probados. Para ello, es necesario conocer cómo se ha desarrollado ese proceso y qué prácticas han hecho posible evaluar el rendimiento del mismo. El benchmarking debe actuar como un mecanismo de cooperación y colaboración entre entidades análogas de cara a compartir información para mejorar sus procesos. (Díez del Valle Medrano & Francisco Torreño, 2016)

(Díez del Valle Medrano & Francisco Torreño, 2016) Mencionan de una forma genérica podemos indicar que a la hora de especificar o definir un benchmark éste debe tener unas características principales:

- Identificar las pruebas a realizar: implica un análisis del sistema, una revisión previa de posibles problemas, así como la identificación de posibles pruebas a implementar.

- Documentar los procesos: supone la reproducción en un diagrama o esquema de los distintos procesos y subprocesos para definir en conjunto su composición.

- Medir los procesos: el éxito de la realización del proceso de benchmark radica en que las medidas efectuadas se hagan con rigor y precisión asegurando que se está midiendo lo mismo y con los mismos criterios.

- Analizar los resultados e identificar las diferencias de rendimiento: con el análisis de los resultados obtenidos se pueden identificar las diferencias entre sistemas.

*Tabla 2. Métricas para medir el rendimiento de las Base de datos*

| **Métrica** | **Definición** |
|---|---|
| **Latencia (ms)** | Tiempo total que tarda en completar la solicitud. |
| **Operaciones\Segundos (Op\S)** | Número de solicitudes que se hacen por segundo. |
| **Errores** | Número de errores que se obtuvieron en el test. |
| **Precio \ Rendimiento (Op \s \ $)** | Cantidad de dinero que cuesta la base de datos en base al rendimiento que se obtiene. se puede medir en operaciones\segundo\precio |

## 2.3 Big Data

El termino “Big Data” suele aplicarse a conjuntos de datos que superan la capacidad del software habitual para ser capturados, gestionados y procesados en un tiempo razonable y por los medios habituales de procesamiento de la información. Dentro del

sector de tecnologías de la formación y la comunicación, big data es una referencia a los sistemas que manipulas grandes conjuntos de datos. (Pérez, 2015)

Big Data puede ser considerada como una tendencia en el avance de la tecnología que ha abierto la puerta a un nuevo enfoque para la comprensión y la toma de decisiones, que se utiliza para describir las enormes cantidades de datos (estructurados, no estructurados y semi- estructurados) que sería demasiado largo y costoso para cargar una base de datos relacional para su análisis. Así, el concepto de Big Data se aplica a toda la información que no puede ser procesada o analizada utilizando herramientas o procesos tradicionales. En términos generales, Big Data y los procesos que dicha técnica representa tiene un amplio espectro de aplicaciones potenciales. (Moreno, 2014)

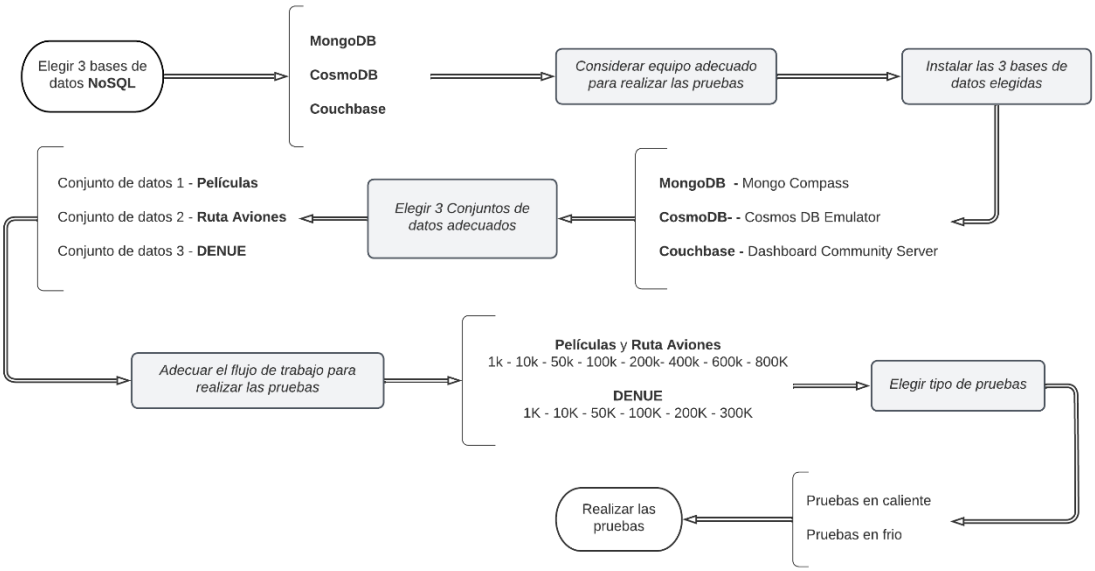
Cada día en el mundo se generan más de 2.5 exabytes de datos. Esto equivale a 1.000.000 de terabytes. La generación de datos no solo crece, explota. El crecimiento exponencial es tan grande, que el 90% de los datos guardados en la actualidad, han sido creados en los dos últimos años. Cada segundo sensores, tabletas, teléfonos y sistemas inteligentes generan cantidades de datos que crecen exponencialmente. Actualmente, la mayoría de los datos almacenados por las nuevas tecnologías no tienen más de dos años. Muchos de estos datos no se procesan porque los sistemas tradicionales de computación no son capaces de procesarlos y muchas empresas no tienen una solución unificada para recogerlos y analizarlos. (Moreno, 2014)

# III. Metodología

Para esta investigación se utilizaron 3 motores de base de datos NoSQL, se eligieron los 3 que se consideraron más adecuados y pertinentes para la misma, estos fueron MongoDB, Azure Cosmo DB y Couchbase. Todos estos motores de base de datos almacenan la información a base de documentos.

Las pruebas se hicieron mediante la interfaz de cada base de datos. En el caso de MongoDB las pruebas se realizaron en MongoDB Compass. En Azure Cosmo DB se realizaron en Azure Cosmos DB Emulator, mientras que en Couchbase se realizaron en el Dashboard de Couchbase Community Server.

*Figura 5. Diagrama de la metodología*




Para la realización de las pruebas era necesario contar con una computadora con características que pudieran soportar el procesamiento de grandes cantidades de información, además de tener una unidad de almacenamiento suficientemente rápida. en este caso se utilizó una laptop Lenovo Legión S7 15ACH6 con los siguientes componentes:

- Procesador: Ryzen 7 5800H
- Ram: 16GB DDR4 3200Mhz
- Almacenamiento: 512 GB SSD 3000 mb\s

Se utilizaron 3 conjuntos de datos diferentes para realizar las pruebas, era importante que estos conjuntos fueran de un tamaño considerablemente grande para poder tener un mejor margen y más posibilidades para realizar estos test e igualmente de esta manera poder simular una situación real, todos estos conjuntos de datos son de origen público:

*Tabla 3. Información de los conjuntos de datos utilizados en esta investigación*

| Nombre del conjunto de datos | Descripción | No. documentos | No. datos por documento | Peso |
|---|---|---|---|---|
| Películas | Datos básicos sobre películas, series, cortos, etc. desde los años 1900 | 1,048,575 | 9 | 321 MB |
| Ruta Aviones | Datos sobre la información de vuelos de aviones en el año 2007 | 474,306 | 30 | 868 MB |
| DENUE | Datos del DENUE de la Ciudad de México del año 2022 hasta el periodo de Mayo | 1,048,575 | 42 | 655 MB |

Se optó por solamente utilizar consultas de lectura en las pruebas, Se repitió 3 veces la misma consulta y se tomó el promedio, se utilizaron 8 cantidades de resultados diferentes para las mediciones. El flujo de trabajo que se utilizó para los conjuntos de datos “Películas” y “Ruta Aviones” fue el siguiente:

1. 1,000 resultados
2. 10,000 resultados
3. 50,000 resultados
4. 100,000 resultados
5. 200,000 resultados
6. 400,000 resultados
7. 600,000 resultados
8. 800,000 resultados

Mientras que para el conjunto de datos “Denue” se utilizaron solamente 6 cantidades diferentes de resultados para las mediciones, esto debido a la cantidad de documentos de este conjunto, ya que es de menor tamaño a los 2 anteriores. El flujo de trabajo fue el siguiente:

1. 1,000 resultados
2. 10,000 resultados
3. 50,000 resultados
4. 100,000 resultados
5. 200,000 resultados
6. 300,000 resultados

La prueba se realizó de manera local, se utilizaron 2 métodos para hacer las pruebas de las bases de datos:

- Test en caliente
    - Las consultas se realizaron de manera consecutiva.
- Test en frío:
    - Las consultas se realizaron apagando el ordenador cada vez que se realizaba una consulta.

El resultado que se intenta conseguir es encontrar cual motor de base de datos NoSQL es el más rápido en cada una de las condiciones, además de encontrar fortalezas y debilidades de estos motores de base de datos comparándolos entre sí. Las consultas que se utilizaron para los test fueron diferentes en cada motor de base de datos, aunque los filtros fueran los mismos, ya que cada motor de base datos tiene su propio lenguaje. Couchbase utiliza el lenguaje N1QL que es un lenguaje muy similar a SQL, CosmoDB utiliza una API SQL para las consultas, mientras que MongoDB utiliza MQL (MongoDB Query Language) que es basado en JavaScript.

# IV. Resultados

Las siguientes tablas y gráficas representan los resultados obtenidos de los test realizados a los motores de base de datos NoSQL MongoDB, CosmoDB, Couchbase, con los distintos conjuntos de datos seleccionados y con los 2 métodos de prueba diferente, estas muestran el promedio de tiempo de ejecución obtenido por cada consulta.

**Conjunto de datos "Películas"**

*Tabla 4. Tiempos promedios recuperados del conjunto de datos Películas en caliente.*

| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 434 ms | 4.66 ms | 30.67 ms |
| 10,000 registros | 487 ms | 36.97 ms | 137.93 ms |
| 50,000 registros | 454 ms | 204.91 ms | 677.53 ms |
| 100,000 registros | 442 ms | 401.58 ms | 1433.33 ms |
| 200,000 registros | 451 ms | 783.42 ms | 2700 ms |
| 400,000 registros | 461 ms | 1357.14 ms | 5133.33 ms |
| 600,000 registros | 664 ms | 2140.56 ms | 7633.33 ms |
| 800,000 registros | 582 ms | 2788.08 ms | 11700 ms |

*Gráfica 1. Tiempos promedios recuperados del conjunto de datos Películas en caliente.*

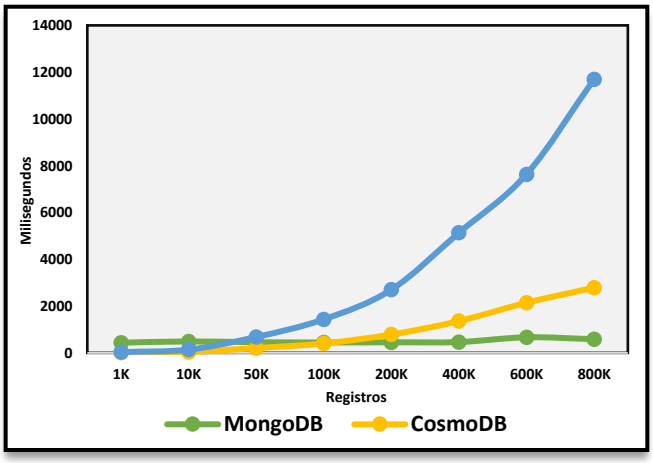
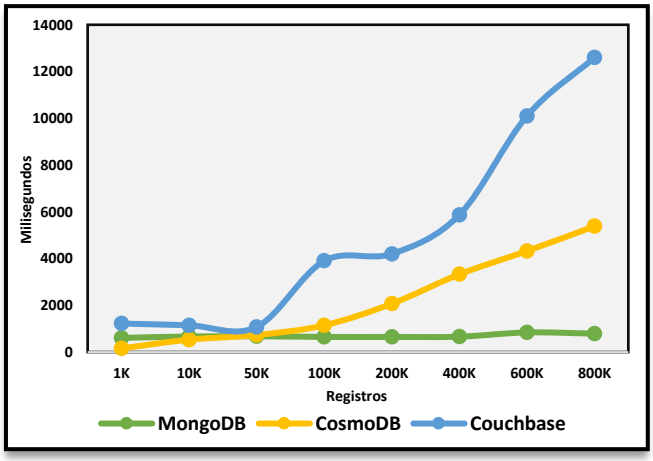



*Tabla 5. Tiempos promedios recuperados del conjunto de datos películas en frío*

| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 603 ms | 157.59 ms | 1220.03 ms |
| 10,000 registros | 666.33 ms | 530.65 ms | 1146.07 ms |
| 50,000 registros | 676.33 ms | 728.79 ms | 1072 ms |
| 100,000 registros | 647.33 ms | 1144.11 ms | 3900 ms |
| 200,000 registros | 646.33 ms | 2067.6 ms | 4200 ms |
| 400,000 registros | 657.67 ms | 3325.03 ms | 5866.67 ms |
| 600,000 registros | 839.33 ms | 4322.43 ms | 10100 ms |

| | | | |
|---|---|---|---|
| 800,000 registros | 788 ms | 5379.47 ms | 12600 ms |

*Gráfica 2. Tiempos promedio recuperados del conjunto de datos películas en frío*




## Conjunto de datos “Ruta aviones”

*Tabla 6. Tiempos promedio recuperados del conjunto de datos Ruta Aviones en caliente*

| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 509 ms | 5.68 ms | 43.77 ms |
| 10,000 registros | 489 ms | 46.36 ms | 306.7 ms |
| 50,000 registros | 766 ms | 262.22 ms | 1633.33 ms |
| 100,000 registros | 545 ms | 409.42 ms | 2966.67 ms |
| 200,000 registros | 554 ms | 901.93 ms | 5833.33 ms |
| 400,000 registros | 516 ms | 1474.01 ms | 11600 ms |
| 600,000 registros | 650 ms | * | * |
| 800,000 registros | 770 ms | * | * |

*Grafica 3. Tiempos promedio recuperados del conjunto de datos Ruta Aviones en caliente*

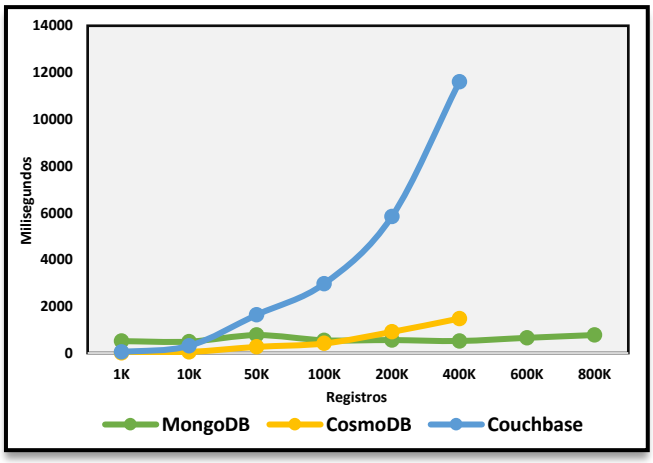
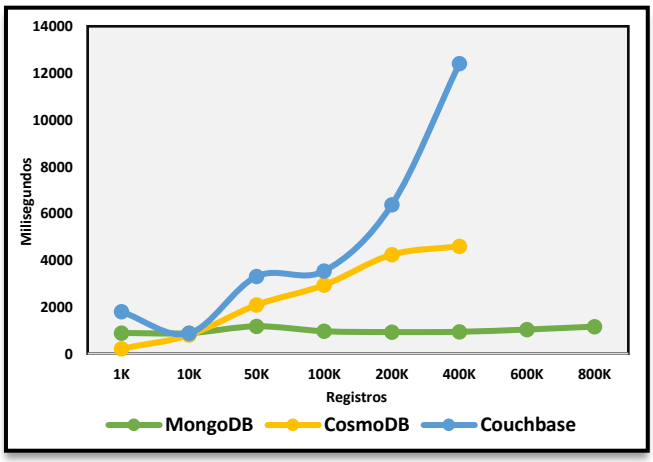



*Tabla 7. Tiempos promedio recuperados del conjunto de datos Ruta Aviones en frio*

| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 882.33 ms | 210.68 ms | 1790.33 ms |
| 10,000 registros | 875.33 ms | 793.07 ms | 865.87 ms |
| 50,000 registros | 1166.67 ms | 2080.67 ms | 3300 ms |
| 100,000 registros | 954.67 ms | 2931.17 ms | 3533.33 ms |
| 200,000 registros | 920.33 ms | 4229.16 ms | 6366.67 ms |
| 400,000 registros | 932.33 ms | 4588.13 ms | 12400 ms |
| 600,000 registros | 1029 ms | * | * |
| 800,000 registros | 1155 ms | * | * |

**En el Conjunto de datos "Ruta Aviones" se tenía previsto que las consultas de 600 mil y 800 mil registros se realizaran correctamente en las 3 bases de datos con los 2 métodos diferentes, sin embargo, en las bases de datos CosmoDB y Couchbase no se pudieron realizar correctamente en ninguno de los métodos, no se conoce certeramente por qué ya que no arrojaba un error en específico, sin embargo, lo más probable es que el problema sea debido a la cantidad de información que arrojan las 2 consultas anteriores ya que podrían ser demasiados datos para el navegador.*

*Gráfica 4. Tiempos promedio recuperados del conjunto de datos Ruta Aviones en frio*




**Conjunto de datos "Denue"**

*Tabla 8. Tiempos promedio recuperados del conjunto de datos Denue en caliente.*

| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 267 ms | 8.49 ms | 57 ms |
| 10,000 registros | 239 ms | 62.99 ms | 409.93 ms |
| 50,000 registros | 268 ms | 280.76 ms | 2033.33 ms |
| 100,000 registros | 275 ms | 588.99 ms | 4600 ms |
| 200,000 registros | 287 ms | 1049.04 ms | 8733.33 ms |
| 300,000 registros | 288 ms | 1514.07 ms | 13000 ms |

*Gráfica 5. Tiempos promedio recuperados del conjunto de datos Denue en caliente.*

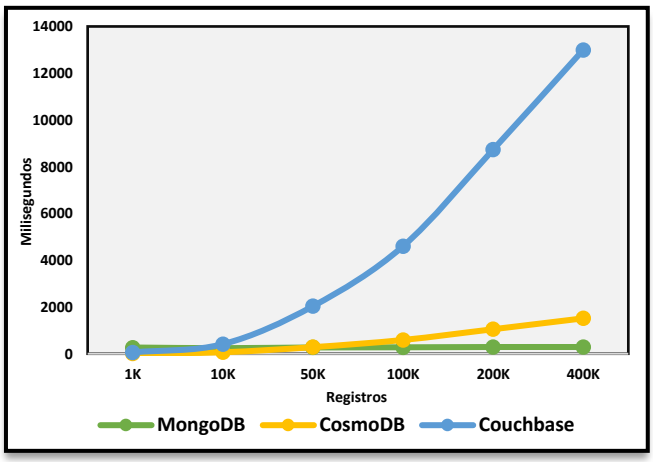
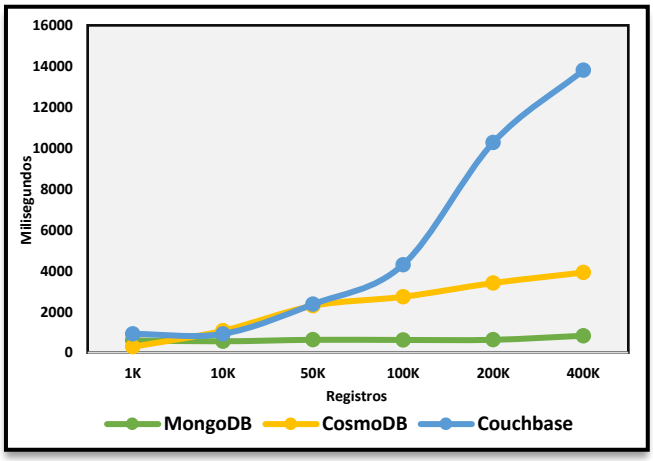



*Tabla 8. Tiempos promedio recuperados del conjunto de datos Denue en frío*

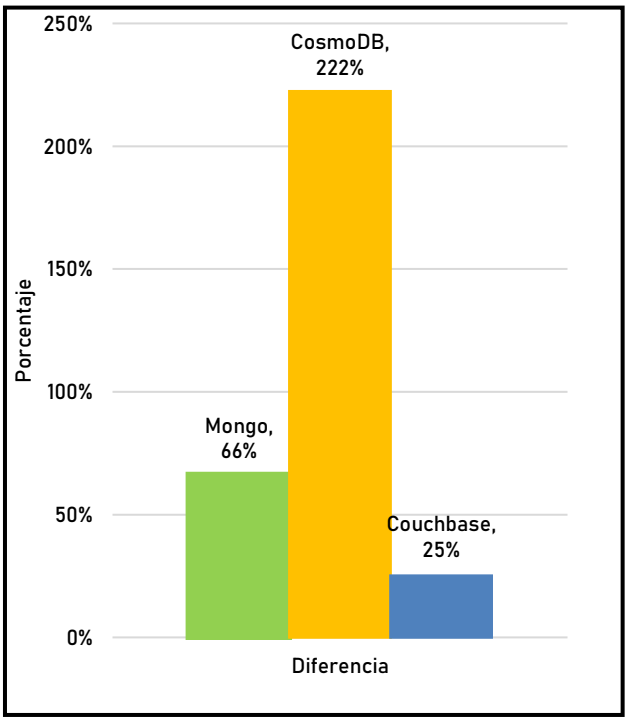
| Registros | MongoDB | CosmoDB | Couchbase |
|---|---|---|---|
| 1,000 registros | 599.67 ms | 282.62 ms | 915.73 ms |
| 10,000 registros | 554.33 ms | 1067.55 ms | 918.3 ms |
| 50,000 registros | 626.67 ms | 2288.93 ms | 2366.67 ms |
| 100,000 registros | 619.33 ms | 2729.98 ms | 4300 ms |
| 200,000 registros | 627.67 ms | 3401.43 ms | 10266.67 ms |
| 300,000 registros | 825 ms | 3917.25 ms | 13800 ms |

*Gráfica 6. Tiempos promedio recuperados del conjunto de datos Denue en frío*

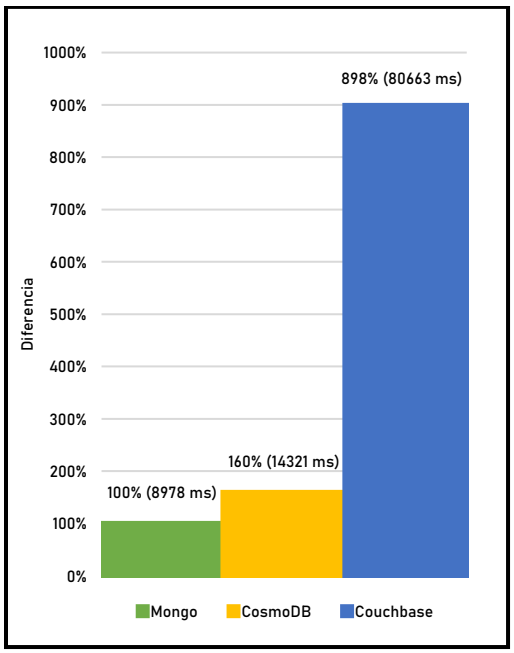



## Graficas adicionales

*Gráfica 7. Porcentaje de la diferencia entre consultas en frío vs caliente*




*Gráfica 8. Porcentaje de la diferencia de la suma de los tiempos arrojados en todos los conjuntos de datos* en caliente entre las 3 bases de datos*




**No se tomaron en cuenta las consultas de 600k ni las de 800k del conjunto de datos "Ruta Aviones" en ninguna base de datos al hacer la suma.*

# V. Conclusión

Al usar bases de datos NoSQL que usan el mismo tipo de dato se elimina una variable que podría afectar a un análisis de los resultados. Entendiendo el hecho que un mejor rendimiento no significa que una base de datos es mejor o peor. Todas las bases de datos testeadas no tuvieron algún problema mayor en todos los procesos (instalación, configuración, subida de datos, consultas) solamente Couchbase tuvo algunas complicaciones menores que tuvieron impacto en el tiempo de desarrollo de esta investigación, sin embargo, una vez que se familiarizo con esta base de datos no se tuvo problema alguno.

La diferencia entre consultas de caliente y frio fueron significativas dependiendo la base de datos, CosmoDB fue el motor de base de datos que perdió más rendimiento comparándolo con su rendimiento en caliente, mientras que Couchbase fue el que menos rendimiento perdió. Couchbase necesito el uso de indexes personalizados para poder tener tiempos competitivos.

En ocasiones esporádicas una consulta con más registros fue más rápida que una con menos registros, en consultas en frio este comportamiento se da por la inestabilidad de este tipo de consulta, es el caso de CosmoDB y Couchbase, sin embargo, en consultas en caliente solo MongoDB mostro ese comportamiento, esto es debido a la dificultad de la consulta.

El resultado que se intentó conseguir era encontrar cual motor de base de datos NoSQL era el más rápido en cada una de las condiciones, MongoDB fue el motor de base de datos con mejor rendimiento, además de encontrar fortalezas y debilidades de estos motores de base de datos comparándolos entre sí, Couchbase tiene más potencial del mostrado en esta investigación, sin embargo es complicado encontrar esa optimización que se desea, CosmoDB Cuenta con el respaldo de Azure-Microsoft por lo que es una fuerte opción si se está familiarizado con alguna tecnología de Microsoft. MongoDB es muy versátil y una de las mejores opciones del mercado.

# Referencias


Aguilar, L. J. (2019). *INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS.* Alfaomega.

Barzu, C. (2017). Estudio del rendimiento de sistemas de gestion de bases de datos. *Universidad Politécnica de Madrid*.

C. Romero, A., G. Sanabria, J., & C. Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21-32.

Díez del Valle Medrano, Á., & Francisco Torreño, D. d. (2016). Comparativa del rendimiento de consultas entre sistemas relacionales (Oracle y MySQL). *E.T.S.I. de Sistemas Informáticos (UPM)*.

Gracia del Busto, H., & Yanes Enríquez, O. (2012). Bases de datos NoSQL. *TELEM@TICA*, 21-33.

J. BERNDT, D., LASA, R., & MCCART, J. (2012). SiteWit Corporation: SQL or NoSQL that is the Question. *University of South Florida*.

Moreno, J. P. (2014). UNA APROXIMACIÓN A BIG DATA. *Revista de Derecho UNE*.

Pérez, M. (2015). *Big Data, Técnicas, herramientas y aplicaciones.* Mexico: Alfaomega.

Tiwari, S. (2011 ). *Professional NoSQL,.* Indianapolis: John Wiley & Sons, Inc.