



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**EVALUACIÓN DE BASE DE DATOS NEWSQL TIDB COMO MEJOR ALTERNATIVA A  
BASES DE DATOS SQL Y NOSQL EN APLICACIONES DE BIG DATA**

**Juan Pablo García Monzón**

Asesorado por el Ing. Sergio Arnaldo Méndez Aguilar

Guatemala, marzo de 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**EVALUACIÓN DE BASE DE DATOS NEWSQL TIDB COMO MEJOR ALTERNATIVA A  
BASES DE DATOS SQL Y NOSQL EN APLICACIONES DE BIG DATA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**JUAN PABLO GARCÍA MONZÓN**

ASESORADO POR EL ING. SERGIO ARNALDO MÉNDEZ AGUILAR

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, MARZO DE 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Carlos Gustavo Alonzo
EXAMINADOR	Ing. Byron Rodolfo Zepeda Arévalo
EXAMINADOR	Ing. Manuel Haroldo Castillo Reyna
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

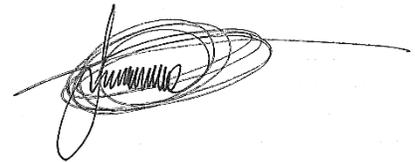


## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **EVALUACIÓN DE BASE DE DATOS NEWSQL TIDB COMO MEJOR ALTERNATIVA A BASES DE DATOS SQL Y NOSQL EN APLICACIONES DE BIG DATA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 9 de febrero de 2022.

A handwritten signature in black ink, appearing to read 'Juan Pablo García Monzón', enclosed within a large, loopy scribble.

**Juan Pablo García Monzón**

Guatemala, 18 de enero de 2023

Ingeniero  
**Carlos Alfredo Azurdía**  
**Coordinador de Privados y Trabajos de Tesis**  
**Escuela de Ingeniería en Ciencias y Sistemas**  
**Facultad de Ingeniería - USAC**

Respetable Ingeniero Azurdía:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **Juan Pablo García Monzón** con carné **201222615** y CUI **2564 97133 0101** titulado **“Evaluación de base de datos NewSQL TiDB como mejor alternativa a bases de datos SQL y NoSQL en aplicaciones de Big Data”** luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fé de que el mismo cumple con los objetivos propuestos en el respectivo protocolo, por consiguiente, procedo a la aprobación correspondiente.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Sergio Arnaldo Méndez Aguilar**  
Colegiado No. 10958



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 19 de enero de 2023

Ingeniero  
**Carlos Gustavo Alonzo**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JUAN PABLO GARCÍA MONZÓN** con carné **201222615** y CUI **2564 97133 0101** titulado “**Evaluación de base de datos NewSQL TiDB como mejor alternativa a bases de datos SQL y NoSQL en aplicaciones de Big Data**”, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

LNG.DIRECTOR.059.EICCSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **EVALUACIÓN DE BASE DE DATOS NEWSQL TIDB COMO MEJOR ALTERNATIVA A BASES DE DATOS SQL Y NOSQL EN APLICACIONES DE BIG DATA**, presentado por: **Juan Pablo García Monzón**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”

Ing. Carlos Gustavo Alonzo  
Director

Escuela de Ingeniería en Ciencias y Sistemas

Msc. Ing. Carlos Gustavo Alonzo  
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, marzo de 2023





Decanato  
Facultad de Ingeniería  
24189101- 24189102  
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.286.2023

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **EVALUACIÓN DE BASE DE DATOS NEWSQL TDB COMO MEJOR ALTERNATIVA A BASES DE DATOS SQL Y NOSQL EN APLICACIONES DE BIG DATA**, presentado por: **Juan Pablo García Monzón**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. Aurelia Anabela Cordova Estrada

Decana



Guatemala, marzo de 2023

AACE/gaoc

## **ACTO QUE DEDICO A:**

### **Mi madre y padre**

Ana Silvia Monzón Monterroso y Luis Enrique García Ocaña, por su amor y apoyo incondicional.

### **Mi hermano**

Enrique Alejandro, por ser una fuente de fraternidad, ayuda y compañerismo.

### **Mi novia**

Clara Marcela García García, por su amor, cariño y ayuda.

### **Mis amigos**

Por su apoyo y hacer mejor las experiencias compartidas.



## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por ser la única universidad pública que me permitió realizar mis estudios de pregrado.
<b>Facultad de Ingeniería</b>	Por proporcionar conocimientos y experiencias.
<b>Mi asesor</b>	Ing. Sergio Arnaldo Méndez Aguilar por su guía, ayuda y apoyo.



# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN .....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN.....	XIX
1. PLANTEAMIENTO.....	1
1.1. Alcance .....	2
1.2. Limites.....	2
1.3. Resultados que se esperan obtener al final del proyecto .....	2
1.4. Estado del arte .....	3
2. CONOCIMIENTO PREVIO .....	7
2.1. Manejo de la Big Data en la actualidad .....	7
2.2. Teorema de CAP para bases de datos SQL y NoSQL .....	9
2.3. Bases de datos relacionales.....	11
2.4. Bases de datos no relacionales.....	11
2.4.1. Bases de datos de documentos .....	12
2.4.2. Almacenes de grafos .....	12
2.4.3. Almacenes de clave-valor.....	12
2.4.4. Orientadas a columnas.....	12
2.5. Bases de datos NewSQL.....	12
2.6. Teorema de CAP para bases de datos NewSQL.....	13
2.7. OLTP .....	13

2.8.	OLAP .....	13
2.9.	HTAP .....	14
2.10.	Comparativa entre SQL, NoSQL y NewSQL.....	14
2.11.	Beneficios de bases de datos NewSQL .....	14
3.	BASES DE DATOS PARA EXPERIMENTACIÓN .....	17
3.1.	MySQL .....	17
3.1.1.	Gestión de Big Data.....	17
3.1.2.	Casos de éxito .....	17
3.1.3.	Servicios administrados en la nube.....	17
3.2.	MongoDB .....	18
3.2.1.	Gestión de Big Data.....	18
3.2.2.	Casos de éxito .....	18
3.2.3.	Servicios administrados en la nube.....	18
3.3.	TiDB .....	19
3.3.1.	Gestión de Big Data.....	19
3.3.2.	Casos de éxito .....	19
3.3.3.	Servicios administrados en la nube.....	19
3.3.4.	Arquitectura distribuida.....	20
4.	PARÁMETROS DE EXPERIMENTO .....	21
4.1.	Herramienta para prueba de base de datos.....	21
4.2.	Experimentos para realizar.....	21
4.3.	Indicadores claves a comparar .....	22
4.4.	Definición de escenarios de pruebas .....	22
4.5.	Características de servidores .....	22
4.5.1.	MySQL.....	22
4.5.2.	MongoDB.....	23
4.5.3.	TiDB.....	23

5.	CONFIGURACIÓN DE INFRAESTRUCTURA PARA IMPLEMENTACIÓN DE EXPERIMENTO .....	25
5.1.	Configuración .....	25
5.2.	Aprovisionamiento de máquinas virtuales con Terraform .....	25
5.3.	Configuración de instalación de software con Ansible .....	29
6.	IMPLEMENTACIÓN DE EXPERIMENTO .....	37
6.1.	Implementación.....	37
6.2.	Configuración de MySQL para implementación de experimento .....	37
6.2.1.	Preparación de experimento de MySQL .....	38
6.2.2.	Ejecución de experimento de MySQL .....	38
6.3.	Configuración de MongoDB para implementación de experimento .....	39
6.3.1.	Preparación de experimento de MongoDB .....	39
6.3.2.	Ejecución de experimento de MongoDB .....	40
6.4.	Configuración de TiDB para implementación de experimento .....	40
6.4.1.	Instalación de la herramienta TiUP .....	41
6.4.2.	Instalar el componente clúster.....	41
6.4.3.	Inicializar el archivo que maneja el clúster .....	41
6.4.4.	Despliegue del clúster .....	42
6.4.5.	Verificar el estado del clúster .....	43
6.4.6.	Iniciar el clúster.....	43
6.4.7.	Creación de base de datos.....	43
6.4.8.	Preparación de experimento para la base de datos TiDB .....	44
6.4.9.	Ejecución de experimento con la base de datos TiDB .....	45

7.	ANÁLISIS DE RESULTADOS .....	47
7.1.	Resultado de MySQL.....	47
7.2.	Resultado de MongoDB.....	48
7.3.	Resultado de TiDB.....	48
7.4.	Análisis de resultados.....	49
	CONCLUSIONES.....	51
	RECOMENDACIONES .....	53
	REFERENCIAS .....	55
	APÉNDICE .....	59
	ANEXOS.....	61

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Ciclo de vida de datos en un ambiente Big Data .....	8
2.	Reporte mundial de gestión servicios de datos.....	8
3.	Interpretación de las tuplas del teorema de CAP.....	9
4.	Instalación de Terraform.....	25
5.	Plantilla de archivo main.tf.....	26
6.	Plantilla de archivo variables.tf .....	26
7.	Plantilla de archivo output.tf.....	28
8.	Archivo provider.tf.....	28
9.	Configuración de cuenta AWS.....	29
10.	Flujo de trabajo de Terraform .....	29
11.	Instalación de Ansible.....	30
12.	Archivo hosts .....	30
13.	Verificación de conexión.....	31
14.	Playbook para la máquina de MySQL.....	32
15.	Playbook para la maquina con MongoDB .....	33
16.	Playbook para la máquina de TiDB.....	34
17.	Playbook para máquinas requeridas por TiDB .....	35
18.	Ejecución de playbooks .....	36
19.	Configuración de base de datos MySQL.....	37
20.	Contenido archivo config .....	38
21.	Preparación de experimento de MySQL .....	38
22.	Ejecución de experimento de MySQL .....	38
23.	Configuración de base de datos MongoDB.....	39

24.	Contenido archivo config .....	39
25.	Preparación de experimento de MongoDB .....	40
26.	Ejecución de experimento de MongoDB .....	40
27.	Instalación de la herramienta TiUP .....	41
28.	Instalación del componente clúster .....	41
29.	Inicializar el archivo cluster.yaml .....	42
30.	Contenido del archivo cluster.yaml.....	42
31.	Despliegue del clúster .....	42
32.	Verificar el estado del clúster .....	43
33.	Iniciar el clúster .....	43
34.	Creación de base de datos.....	44
35.	Contenido archivo config .....	44
36.	Preparación de experimento de TiDB .....	44
37.	Ejecución de experimento de TiDB .....	45

## TABLAS

I.	Comparación entre las bases de datos SQL, NoSQL y NewSQL .....	14
II.	Parámetros para experimentos con sysbench .....	21
III.	Especificaciones de servidor MySQL.....	23
IV.	Especificaciones de servidor MongoDB.....	23
V.	Especificaciones de servidor TiDB.....	24
VI.	Explicación de variables .....	27
VII.	Variables .....	31
VIII.	Paquetes para instalar en máquina de MySQL.....	32
IX.	Paquetes para máquina de MongoDB .....	33
X.	Paquetes para instalar en máquina principal de TiDB .....	34
XI.	Paquetes para instalar en máquinas requeridas por TiDB .....	34
XII.	Módulos y opciones usados en playbooks.....	35

XIII.	Resultados de experimento de MySQL.....	47
XIV.	Promedio de datos de experimento de MySQL .....	47
XV.	Resultados de experimento de MongoDB.....	48
XVI.	Promedio de datos de experimento MongoDB .....	48
XVII.	Resultados de experimento de TiDB.....	49
XVIII.	Promedio de datos de experimento de TiDB .....	49



## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>GB</b>	Gigabyte
<b>mseg</b>	Milisegundos
<b>seg</b>	Segundos



## GLOSARIO

<b>ACID</b>	Por sus siglas en inglés, Atomicidad, Coherencia, Aislamiento y Durabilidad.
<b>AWS</b>	Nube pública de Amazon.
<b>BASE</b>	Por sus siglas en inglés, Básicamente disponible, estado blando, coherencia eventual.
<b>CAP</b>	Por sus siglas en inglés, Consistencia, Disponibilidad, Tolerancia a la partición.
<b>Clúster</b>	Grupos de servidores que se gestionan juntos y participan en la gestión de carga de trabajo.
<b>DaaS</b>	Por sus siglas en inglés, Bases de datos como Servicio.
<b>EC2</b>	Servicio de AWS para la creación de servidores.
<b>IaC</b>	Por sus siglas en inglés, Infraestructura como Código.
<b>IP</b>	Por sus siglas en inglés, Protocolo de Internet.
<b>JSON</b>	Es un formato ligero de intercambio de datos.

<b>MariaDB</b>	Base de datos relacional.
<b>MongoDB</b>	Base de datos no relacional.
<b>MySQL</b>	Base de datos relacional.
<b>OLAP</b>	Por sus siglas en inglés, Procesamiento Analítico en Línea.
<b>OLTP</b>	Por sus siglas en inglés, Procesamiento de Transacciones en Línea.
<b><i>Playbook</i></b>	Son los pasos para realizar. pueden incluir una o más tareas, que se realizan de manera ordenada y una al mismo tiempo.
<b><i>Pod</i></b>	Grupo de máquinas empaquetadas en un solo ambiente de desarrollo.
<b>RDBMS</b>	Por sus siglas en inglés, Sistema de Administración de Bases de Datos Relacionales.
<b>SQL</b>	Lenguaje de programación estandarizado que se utiliza para administrar bases de datos relacionales.
<b><i>Stateless</i></b>	Sistema que no almacena información sobre operaciones anteriores ni se hace referencia a ellas.
<b>TIC</b>	Tecnologías de la Información y la Comunicación.

<b>TiDB</b>	Base de datos NewSQL.
<b>TiKV</b>	Es un motor de almacenamiento transaccional distribuido de clave-valor.
<b>TiUP</b>	Es un componente de tipo administrador para TiDB.
<b>WSL2</b>	Programa para poder correr Linux de forma nativa en Windows.
<b>XML</b>	Lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.



## **RESUMEN**

Las bases de datos son herramientas que permiten registrar, clasificar, mantener y consultar, de manera sistemática, los datos que un proyecto o institución produce para luego convertirlos en información útil.

En este documento se realiza una comparación entre los tres conceptos de bases de datos que existen, y se evalúa cual puede ser una mejor opción para el manejo de una cantidad de datos de tipo Big Data. Utilizando específicamente para SQL la base de datos, MySQL, para la base de datos no relacional, MongoDB, y para la base de datos NewSQL, TiDB.



# OBJETIVOS

## General

Evaluar la base de datos NewSQL TiDB como mejor alternativa a bases de datos SQL y NoSQL en aplicaciones de Big Data.

## Específicos

1. Desarrollar un estado del arte con las últimas investigaciones relevantes sobre el origen y estudio de las bases de datos NewSQL.
2. Listar y describir las ventajas de las bases de datos NewSQL en comparación de las bases de datos SQL y NoSQL.
3. Comparar la base de datos NewSQL TiDB contra la SQL MySQL y la NoSQL MongoDB utilizando un banco de datos de tipo Big Data detallando pros y contras de cada una.
4. Identificar posibles casos de aplicación en las cuales las bases de datos NewSQL ofrecen mejores ventajas que las bases de datos SQL y NoSQL.
5. Definir los aspectos clave de comparación entre bases de datos NewSQL, SQL y NoSQL, ayudando al lector a tener una idea clara de estas tecnologías.



## INTRODUCCIÓN

Este proyecto de graduación consistirá en demostrar, a través de la experimentación, los beneficios de una base de datos NewSQL a comparación de SQL y NoSQL. Evaluando distintos aspectos como consistencia de información, velocidad de procesamiento para Big Data, consumo de recursos, procesamiento en tiempo real, entre otros (Zhou & Su, s.f).

En este trabajo se evaluará a TiDB como tecnología NewSQL, ya que es compatible con MySQL (Tocker, s.f.) lo cual brinda una base de comparación por sus características similares, con relación a bases de datos instaladas localmente o algunos servicios en la nube como los mencionados anteriormente.

Además de esto la experimentación nos brindará resultados, mismos que nos ayudarán a analizar y dar una opinión objetiva acerca de estas tecnologías de bases de datos, para que la persona que lea este documento tenga una base de conocimiento para una elección en base a su propia perspectiva y sus propias necesidades.



## 1. PLANTEAMIENTO

Con el rápido desarrollo de las TIC se ha tenido un incremento sin precedentes de la información que la humanidad produce, llegando incluso a la conclusión que el 90 % de la documentación mundial se ha creado en estos últimos años (Marr, s.f).

Este gran acontecimiento también es un reto para todos los servidores que proveen y reciben información, poniendo en estrés las infraestructuras para contener estos datos; así que venimos de un paradigma de datos estructurados, que aunque sigue siendo de los sistemas de administración de bases de datos más utilizados (statista, s.f) , a utilizar el paradigma de bases de datos no relacionales donde las relaciones no son necesarias y por ende es mucho más veloz la capacidad de consulta y despliegue de datos.

Alrededor del año 2011 (Goldberg, s.f) empezó a surgir otro concepto donde se hizo una pregunta interesante, ¿Qué pasa si juntamos lo mejor de 2 mundos?, naciendo entonces el paradigma de NewSQL mezclando lo relacional con lo no relacional trayendo a la mesa una forma más rápida, eficiente y eficaz del tratado de datos.

Por lo que la vida de este proyecto de tesis parte del núcleo de una pregunta sencilla que se ramifica en algo más complejo, ¿Qué paradigma es el mejor, o mejor dicho cuál paradigma tiene un mejor tratado de datos?, este trabajo tiene la intención de hacer una comparativa entre estos conceptos a base de diferentes experimentos y a partir de esto analizar los resultados y contribuir con una

respuesta objetiva para que el lector pueda elegir y aplicar la solución dependiendo de sus necesidades.

### **1.1. Alcance**

El alcance de este proyecto de tesis será realizar una investigación teórica acerca de las diferentes bases de datos SQL, NoSQL y NewSQL, centrándose en la base de datos TiDB al mostrar sus beneficios al tratar con un banco de datos de tipo Big Data además de comprobar su rendimiento.

### **1.2. Limites**

Esta investigación utilizará las herramientas TiDB, MySQL y MongoDB, para realizar los experimentos de comparación entre los diferentes paradigmas con base en el teorema de CAP.

### **1.3. Resultados que se esperan obtener al final del proyecto**

Se tiene la proyección de crear un documento que tenga la función tanto de explicar la teoría de las bases de datos NewSQL, pero también sus beneficios al usarlo con una cantidad de datos de tipo Big Data, además de una serie de experimentos y de procesos que se pueden realizar con la base de datos TiDB que pueda ser de ayuda para un usuario con un conocimiento previo de bases de datos SQL y NoSQL.

#### **1.4. Estado del arte**

Jonatan Röö, Mathias Johansson, 2020, Performance comparison between NewSQL and SQL Sharded TiDB vs MariaDB, investigación teórica, estudiantes y profesionales de Ciencias de la Computación.

El objetivo general es reunir conocimientos sobre el dominio e investigar el área del problema y encontrar métodos que puedan probar diferentes aspectos del rendimiento.

Los objetivos específicos son reunir información acerca de bases de datos puras, comprobar como TiDB se compara con MariaDB en términos de rendimiento.

Recientemente, han aparecido bases de datos NewSQL que combinan las características NoSQL con la compatibilidad con SQL y cumplimiento de ACID que normalmente no se encuentra en las bases de datos NoSQL. Debido a su reciente aparición, existe un vacío de conocimiento en la literatura sobre las bases de datos NewSQL. Por lo tanto, este trabajo compara el rendimiento de TiDB con MariaDB en una prueba de rendimiento.

Se relaciona con mi proyecto de tesis ya que realiza una comparación entre TiDB, además de la base teórica el tener ya un respaldo más sólido acerca de esta herramienta reciente me sirve para tener una mejor guía para mi desarrollo.

Khasawneh, Tariq, AL-Sahlee, Mahmoud, Safia, Ali, 2020, SQL, NewSQL, and NoSQL Databases: A Comparative Survey, artículo científico, investigación teórica, estudiantes y profesionales de Ciencias de la Computación.

Comparar las diferentes bases de datos como NewSQL, NoSQL y SQL basándose en criterios diversos como el teorema de CAP y propiedades BASE.

Este trabajo tiene como objetivo proporcionar una visión general de los diferentes sistemas de gestión NoSQL, clasificándolos en cuatro categorías principales: almacenes de valores clave, orientados a columnas, orientados a documentos y orientados a gráficos, comparando cada una de las categorías utilizando múltiples criterios incluyendo el teorema CAP y las propiedades BASE.

Se relaciona con mi proyecto de tesis ya que uno de sus criterios más importantes es utilizar el teorema de CAP concepto que me ayudará a comparar TiDB, MySQL y MongoDB.

Raj, Pethuru, 2018, A Detailed Analysis of NoSQL and NewSQL Databases for Big data Analytics and Distributed Computing, investigación teórica, estudiantes y profesionales de Ciencias de la Computación.

El objetivo general es hacer una inmersión profunda en bases de datos NoSQL sus usos y aplicaciones.

El objetivo específico es realizar un análisis detallado de bases de datos NoSQL y NewSQL para analíticas de tipo Big Data y Computación distribuida.

Principalmente tiene el objetivo de contar todo sobre las diversas bases de datos NoSQL y NewSQL y cómo y cómo resultan útiles para aumentar, acelerar

y automatizar el complicado fenómeno de la de la próxima generación de análisis de datos.

Se relaciona con mi proyecto de tesis ya que proporciona una base teórica sólida acerca de los conceptos de bases de datos y específicamente en NewSQL y NoSQL, además de relacionarlo con analíticas de tipo Big Data.

Kaur, Karambir, 2017, Performance evaluation of NewSQL Databases, investigación teórica, artículo científico, estudiantes y profesionales de Ciencias de la Computación.

El objetivo es comentar y describir acerca de varias bases de datos NewSQL, centrándose en sus beneficios, características en OLTP para Big Data.

El fin de este trabajo es proporcionar la lista de bases de datos NewSQL populares en tablas categorizadas. Este trabajo de tesis abarca principalmente la comparación de la evaluación entre cuatro bases de datos NewSQL: NuoDB, VoltDB, MemSQL, y Cockroach DB en base a varios parámetros como la latencia de lectura, latencia de escritura, latencia de actualización y tiempo de ejecución.

Se relaciona con mi proyecto de tesis ya que proporciona una comparación entre varias bases de datos NewSQL y que cuenta con datos profundos acerca de cómo funcionan algunas de las bases de datos NewSQL.

Barzu, Claudiu, 2017, Estudio del rendimiento de sistemas de gestión de bases de datos NewSQL, investigación teórica, estudiantes y profesionales de Ciencias de la Computación.

El objetivo general del trabajo se puede definir como la realización de un estudio acerca del rendimiento de varios sistemas de bases de datos no relacionales.

Los objetivos específicos son el estudio de arquitectura de Apache Phoenix y Splice Machine, indicando las principales diferencias entre ambos sistemas, las ventajas e inconvenientes de cada herramienta y casos de uso idóneos para cada uno además de definir un procedimiento teórico que cubra el análisis de rendimiento y escalabilidad de los sistemas estudiados, realizar un benchmarking de los sistemas en base a la especificación teórica definida, realizar una comparación entre ambos sistemas, analizando las diferencias entre los tiempos de respuesta y el número de operaciones máximo por segundo obtenido.

Se relaciona con mi proyecto de tesis ya que realiza una comparación entre dos sistemas de bases de datos, que, aunque no son los que yo comparare si me ayuda para tomar en cuenta ciertos indicadores o parámetros de rendimiento como latencia de escritura y lectura, escalabilidad, rendimiento a la hora de hacer varias operaciones, entre otras.

## 2. CONOCIMIENTO PREVIO

### 2.1. Manejo de la Big Data en la actualidad

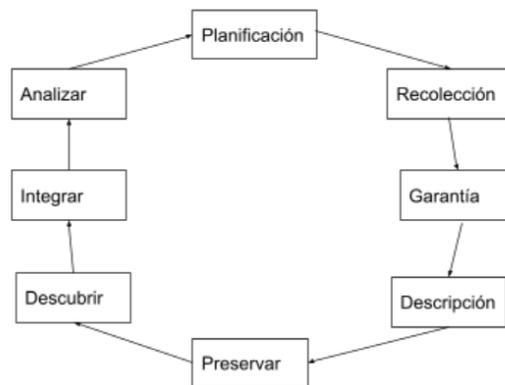
Se puede comenzar desentrañando de ¿Qué es en realidad Big Data?, y se estaría a la merced de una avalancha de términos y jerga que complica en vez de simplificar el entendimiento de este concepto, ya que su traducción es dato grande, una definición ambigua. Por lo que la primera capa de dificultad sería explicar que es Big Data, según Walter Sosa Escudero se refiere al "volumen y tipo de datos provenientes de la interacción con dispositivos interconectados, teléfonos celulares, tarjetas de crédito, cajeros automáticos, relojes inteligentes, computadoras personales, dispositivos GPS y cualquier objeto capaz de producir información y enviarla electrónicamente a otra parte" (Sosa Escudero, s.f, p. 20).

Con esta base de conocimiento pasamos a otro desafío y es que, obviamente como son datos masivos, su conducta es anárquica y espontánea ya que los datos que generamos ya no solo son para llenar un formulario, encuesta o pago de servicios sino también para contar los pasos cuando caminamos, una ruta con menos tráfico o notificaciones de un nuevo episodio de nuestra serie favorita. Una técnica para ubicar los puntos más importantes es con las 4Vs de la Big Data: Volumen, Velocidad, Variedad, Veracidad (Sosa Escudero, s.f, p. 21).

El manejo de datos de tipo Big Data usa analíticas en tiempo real; desde el caso donde se analiza el estado actual de un proyecto y cuál será su proyección en unos años, o ver la tendencia de una población a partir de un suceso anormal. Tomando a la pandemia del SarsCov-2 como ejemplo y como con Google

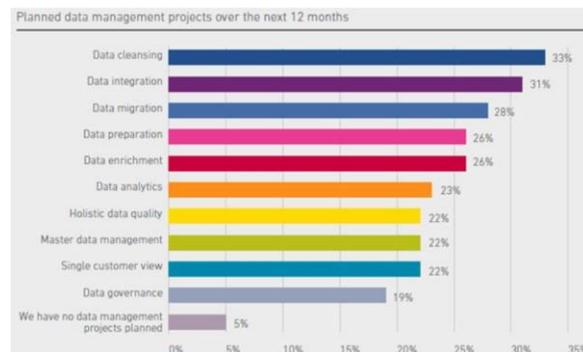
Analytics se ha mantenido información de forma diaria. Todos estos servicios son a partir de una gestión de datos que se pueden comprimir en este diagrama de ciclo de vida:

Figura 1. **Ciclo de vida de datos en un ambiente Big Data**



Fuente: elaboración propia.

Figura 2. **Reporte mundial de gestión servicios de datos**



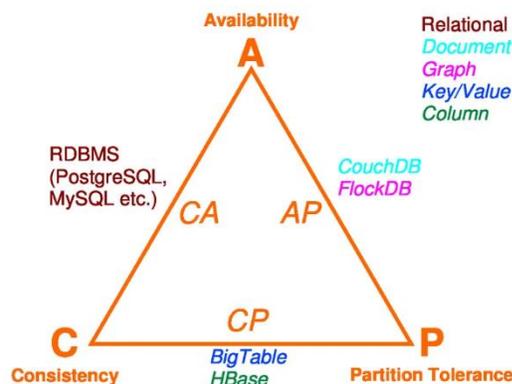
Fuente: Harvey & Ehrlich. (2017). *Planned data management projects*. Consultado el 15 de febrero de 2022. Recuperado de <https://www.datamation.com/big-data/big-data-management>.

## 2.2. Teorema de CAP para bases de datos SQL y NoSQL

Para estas bases de datos los 3 puntos del teorema de CAP no se cumplen a la vez, o sea, no se puede tener consistencia y disponibilidad para actualizaciones al mismo tiempo que el sistema esté en el proceso de una partición. Para estos casos están categorizados en las siguientes tuplas:

- CP: se garantiza la consistencia de los datos entre los diferentes nodos y la partición se tolera, se sacrifica la disponibilidad provocando que el sistema pueda fallar o tenga una respuesta lenta a una consulta.
- AP: se garantiza el acceso y el sistema es capaz de tolerar la partición de los nodos, se sacrifica la consistencia de datos provocando que no se repliquen los valores de los datos en diferentes nodos al instante.
- CA: se garantiza el acceso a la información y la consistencia de datos en diferentes réplicas, se sacrifica la partición de los nodos provocando que no sea soportada por el sistema de forma simultánea en los nodos.

Figura 3. Interpretación de las tuplas del teorema de CAP



Fuente: Researchgate. (2014). *Teorea CAP*. Consultado el 15 de febrero de 2022.

Recuperado de <https://www.researchgate.net/figure/Figura-52->

Teorema-CAP-obtenido-de-56\_fig7\_311406418.

Las bases de datos SQL otorgan mayor relevancia a la Consistencia y la Disponibilidad. Si ocurre un fallo de comunicación entre los nodos, habrá problemas en el sistema generando errores al procesar una solicitud.

En cambio, las bases de datos NoSQL otorgan más peso al rendimiento de procesamiento de una gran cantidad de transacciones de distintas fuentes, permitiendo la escalabilidad horizontal de los recursos, además de que la tolerancia a las particiones de los nodos dentro de una red es soportada.

#### Ejemplos de bases de datos CA SQL

- Oracle
- MariaDB
- MySQL

#### Ejemplos de bases de datos CP NoSQL

- MongoDB
- Redis
- Big Table

#### Ejemplos de bases de datos AP NoSQL

- CassandraDB
- DynamoDB
- CouchDB

### **2.3. Bases de datos relacionales**

“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.” (AWS, s.f).

Lo más importante de este concepto es su relación con ACID un estándar que las transacciones deben de cumplir para asegurar una relación exitosa:

- Atomicidad: se debe ejecutar toda la transacción, si una parte esta errónea toda falla.
- Consistencia: los datos en la transacción deben cumplir las reglas definidas.
- Aislamiento: cada transacción es independiente.
- Durabilidad: todos los cambios realizados por una transacción son permanentes.

Entre los ejemplos más comunes tenemos PostgreSQL, Oracle, MySQL, MariaDB, Aurora DB.

### **2.4. Bases de datos no relacionales**

No necesita del estándar ACID para poder realizar una transacción, esta falta de relación permite que la base de datos que administre sea escalable y ágil en la consulta y escritura de nuevos registros. Existen 4 tipos.

### **2.4.1. Bases de datos de documentos**

Almacenan información en registros, cada uno funciona como una unidad autónoma de información, utiliza documentos para el almacén de los registros y los datos asociados a ellos. Los documentos más utilizados son en formato JSON o XML. Algunos ejemplos son MongoDB, DynamoDB y Azure Cosmos.

### **2.4.2. Almacenes de grafos**

Utiliza grafos para representar los datos interconectados y almacenar información sobre las redes de datos, como las conexiones sociales. Algunos ejemplos son Neo4J y Giraph.

### **2.4.3. Almacenes de clave-valor**

Almacena cada elemento en un formato de clave-valor donde el nombre del atributo es la clave y su valor. Algunos ejemplos son Redis, BerkeleyDB, Riak.

### **2.4.4. Orientadas a columnas**

Almacenan los datos en columnas y tienen la posibilidad de realizar consultas en grandes conjuntos de datos. Algunos ejemplos son CassandraDB, HBase.

## **2.5. Bases de datos NewSQL**

Combinan los 2 paradigmas anteriores así que proporciona la habilidad de poder manejar grandes cantidades de datos con escalabilidad y agilidad, pero utiliza el estándar ACID para poder relacionar los datos.

Por lo que varias empresas y entusiastas de administración de datos tipo Big Data empiezan a utilizarlo como una alternativa para poder manejar de una forma más consistente la analítica en tiempo real para una toma de decisiones más segura.

## **2.6. Teorema de CAP para bases de datos NewSQL**

El Teorema CAP mencionado anteriormente para las bases de datos relacionales y no relacionales tenían la característica que solo se podían elegir 2 atributos y se listaron varios casos para las diferentes combinaciones, pero con el concepto de NewSQL se puedan cumplir los 3 atributos. Algunos de los ejemplos que cumplen este hecho son: TiDB, ClustrixDB, CockroachDB, NuoDB, TokuDB, CosmosDB (Cupas, s.f).

## **2.7. OLTP**

Es un tipo de procesamiento de información que consiste en ejecutar una cantidad de transacciones ocurriendo de forma simultánea (Oracle, s.f). Aunque el origen de su creación fue para transacciones financieras, ahora el concepto de transferencia es tan amplio que ha provocado que pueda interactuar con otras situaciones. Por lo que OLTP tiene ahora la capacidad de realizar la ejecución de grandes cantidades de transferencias en tiempo real por muchas personas a la vez.

## **2.8. OLAP**

Es un tipo de procesamiento de información que consiste en analizar un banco de información que se encuentra a disposición (Oracle, s.f), es una herramienta utilizada con frecuencia en ambientes relacionados a business

intelligence ya que permite calcular situaciones complejas y predicciones de escenarios.

## 2.9. HTAP

Mezcla las mejores características de los procesamientos anteriores permitiendo un análisis en tiempo real de una base de transacciones sin afectar la dinámica de estas (Hydra, s.f).

## 2.10. Comparativa entre SQL, NoSQL y NewSQL

Luego de describir los paradigmas de base de datos, se empezará por comparar las características más importantes.

Tabla I. **Comparación entre las bases de datos SQL, NoSQL y NewSQL**

<b>Característica</b>	<b>SQL</b>	<b>NoSQL</b>	<b>NewSQL</b>
Relacional	Si	No	Si
ACID	SI	No	Si
SQL	SI	No	Si
OLTP	SI	No	Si
CAP	CA	CP, AP	CAP

Fuente: elaboración propia.

## 2.11. Beneficios de bases de datos NewSQL

Entre las características de la solución de NewSQL que se pueden considerar como beneficios para nuestros sistemas de administración podemos mencionar que:

- Utiliza el lenguaje SQL como interacción principal con la base de datos, facilitando la interfaz del administrador con el sistema.
- Tiene la capacidad de utilizar el estándar ACID para transacciones.
- Utiliza un mecanismo de control non-locking simultáneo el cual es útil para la lectura en tiempo real sin crear conflictos con la escritura.
- Es capaz de manejar una arquitectura escalable, paralela y shared-nothing que pueda ser utilizada por una gran cantidad de nodos sin sufrir cuellos de botella.
- Es más rápida que los OLTP RDBMS tradicionales.
- Posee la capacidad de brindar un servicio distribuido.



## **3. BASES DE DATOS PARA EXPERIMENTACIÓN**

### **3.1. MySQL**

Exploraremos tres puntos importantes de MySQL, entre ellos su gestión para Big Data, casos de éxito de diferentes empresas o instituciones en el manejo de Big Data y por último que servicios presta en las diferentes nubes públicas.

#### **3.1.1. Gestión de Big Data**

Aunque MySQL no es regularmente asociada al manejo de Big Data está el motor de almacenamiento InnoDB (Vileikis, s.f), aunque es tediosa de configurar, por lo que muchos ambientes empresariales o de otra índole no la toman en cuenta.

#### **3.1.2. Casos de éxito**

La utilizan diferentes empresas como Booking.com, Github, Youtube, Spotify por mencionar a los principales proveedores (MySQL, s.f).

#### **3.1.3. Servicios administrados en la nube**

Destaca por su velocidad, estabilidad, facilidad de uso y de código abierto (Geekflare, s.f). Se encuentra en las nubes públicas principales.

## **3.2. MongoDB**

Exploraremos tres puntos importantes de MongoDB, entre ellos su gestión para Big Data, casos de éxito de diferentes empresas o instituciones en el manejo de Big Data y por último que servicios presta en las diferentes nubes públicas.

### **3.2.1. Gestión de Big Data**

Su administración de Big Data es con la herramienta MongoDB Atlas una aplicación de manejo de datos utilizando como motor a MongoDB, esta cuenta con herramientas como análisis en tiempo real, estadísticas, copias de seguridad continuas, resiliencia entre regiones, entre otros servicios.

### **3.2.2. Casos de éxito**

Los casos de éxito más sonados son: Google, EA, SEGA. Esto del lado del entretenimiento y medios de comunicación, en otras industrias como la de servicios médicos se encuentra Humana y AstraZeneca (MongoDB, s.f).

### **3.2.3. Servicios administrados en la nube**

Ofrece un servicio DaaS llamado Mongo Atlas, es utilizado como un clúster listo para ser usado. Esta en las nubes públicas principales (MongoDB, s.f).

### **3.3. TiDB**

Exploraremos tres puntos importantes de TiDB, entre ellos su gestión para Big Data, casos de éxito de diferentes empresas o instituciones en el manejo de Big Data y por último que servicios presta en las diferentes nubes públicas.

#### **3.3.1. Gestión de Big Data**

Al ser una base de datos HTAP provee un análisis en tiempo real y servicios de data *Warehouse*. Además de localizar datos que tienen una alta concurrencia, también provee a gran escala servicios y consultas de *business intelligence*. Estas características permiten que se utilice para construir un sistema de análisis de datos en tiempo real. También permite integrarse a herramientas de Big Data como Apache *Flink*.

#### **3.3.2. Casos de éxito**

Resalta en el mercado asiático, algunos de los clientes más importantes son Xiaomi, Lenovo, WeBank, TCL, Bilibili, y Hulu (Huang, s.f).

#### **3.3.3. Servicios administrados en la nube**

Cuenta con la solución TiDB Cloud donde lo único que se necesita para utilizarla es registrar un usuario, a partir de ese momento es posible manejar un clúster con esta tecnología. Esto ayuda a facilitar el interactuar con TiDB. Recientemente hicieron alianza con AWS y GCP para que TiDB aparezca en el marketplace de estas nubes públicas (Liu, s.f).

### 3.3.4. Arquitectura distribuida

Para que TiDB pueda correr el clúster más básico se necesita de 6 máquinas, las cuales son:

- TiUP: se instalará TiUP, para administrar la arquitectura.
- TiDB: Recibe peticiones SQL.
- TiKV: se encarga de almacenar los datos. TiKV
- Placement Driver: almacena los metadatos de la distribución de datos en tiempo real de cada nodo TiKV y la estructura de la topología de todo el clúster TiDB, proporciona la interfaz de usuario de gestión de TiDB Dashboard y asigna ID de transacciones a las transacciones distribuidas.
- Monitoring: se utiliza Prometheus para almacenar la información del monitoreo del clúster.
- Grafana: se utiliza Grafana para visualizar la información recabada por Prometheus.

## 4. PARÁMETROS DE EXPERIMENTO

### 4.1. Herramienta para prueba de base de datos

La herramienta para utilizar es sysbench, su repositorio nos da una breve explicación de que realiza “Herramienta de benchmark multihilo basada en LuaJIT. Se utiliza con mayor frecuencia para los puntos de referencia de la base de datos” (Kopytov, s.f).

### 4.2. Experimentos para realizar

Los experimentos consistirán en poner a prueba, bajo los mismos parámetros, la creación, inserción y el manejo de hilos de cada base de datos.

Tabla II. **Parámetros para experimentos con sysbench**

<b>Parámetro</b>	<b>Valor</b>
Registros	500 000
Tablas	8
Tiempo	120 seg
Hilos	128
Intervalo de reporte	10 seg

Fuente: elaboración propia.

### **4.3. Indicadores claves a comparar**

- Latencia media del experimento: proporciona la media de los valores de latencia que se dieron con cada evento en el experimento, su dimensional es en microsegundos.
- Tiempo total del experimento: proporciona el tiempo total que duro el experimento, su dimensional es en segundos.
- Número total de eventos en el experimento: proporciona la cantidad de eventos que sucedieron a lo largo de la duración total del experimento.

### **4.4. Definición de escenarios de pruebas**

Se tendrán servidores virtuales en AWS, para correr las bases de datos respectivas. El sistema operativo de los servidores debe ser una distribución de Linux ya que la herramienta sysbench corre de mejor forma en este ambiente la distribución será basada en Debian específicamente Ubuntu 18.04 LTS.

### **4.5. Características de servidores**

Para poder llevar a cabo de forma exitosa los diferentes experimentos a realizar en las bases de datos, utilizaremos servidores virtuales en AWS todos con las mismas características de Software y Hardware. A continuación, se presentan estas características.

#### **4.5.1. MySQL**

En el caso de MySQL solo se utilizará un servidor con las siguientes especificaciones.

Tabla III. **Especificaciones de servidor MySQL**

<b>Especificación</b>	<b>Valor</b>
Maquinas	1
CPU	t2. medium
RAM	4 GB
ROM	8 GB
Software	Ubuntu 18.04 LTS

Fuente: elaboración propia.

#### **4.5.2. MongoDB**

En el caso de MongoDB solo se utilizará un servidor con las siguientes especificaciones.

Tabla IV. **Especificaciones de servidor MongoDB**

<b>Especificación</b>	<b>Valor</b>
Maquinas	1
Instancia	t2. medium
RAM	4 GB
ROM	8 GB
Software	Ubuntu 18.04 LTS

Fuente: elaboración propia.

#### **4.5.3. TiDB**

La razón de que se usen 6 máquinas es que necesitamos 1 para la administración como las bases de datos anteriores, y otras 5 que se utilizan en el clúster que necesita TiDB para poder correr su estructura

distribuida, en su arquitectura se encuentra *TiDB*, *TiKV*, *Grafana*, *pd* y monitoreo.

Tabla V. **Especificaciones de servidor TiDB**

<b>Especificación</b>	<b>Valor</b>
Maquinas	6
Instancia	t2. Médium
RAM	4 GB
ROM	8 GB
Software	Ubuntu 18.04 LTS

Fuente: elaboración propia.

## 5. CONFIGURACIÓN DE INFRAESTRUCTURA PARA IMPLEMENTACIÓN DE EXPERIMENTO

### 5.1. Configuración

Para la configuración de la infraestructura se usarán los parámetros explicados en el capítulo anterior, y herramientas laC.

El uso de las herramientas laC se ejecutarán en un servidor local con Windows 10 con WSL2 con Ubuntu 18.04 LTS instalado. Los archivos de la implementación del experimento estarán en el repositorio proyecto\_tesis.

### 5.2. Aprovisionamiento de máquinas virtuales con Terraform

Posterior a la instalación de Terraform se creará un directorio de configuración que alojará las carpetas, terraform y ansible en nuestra máquina local.

Figura 4. Instalación de Terraform

```
$ mkdir nombre_directorio
$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl
$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
$ sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
$ sudo apt-get update && sudo apt-get install terraform
```

Fuente: elaboración propia.

En el directorio terraform se creará una serie de archivos comenzando con main.tf, para configurar declaraciones que manejen el servicio EC2 de AWS. Los

valores de los atributos en el archivo son una dependencia implícita de las variables configuradas en el archivo, variables.tf. El recurso utilizado es aws\_instance el cual nos permite manejar los atributos de una instancia.

Figura 5. **Plantilla de archivo main.tf**

```
resource "aws_instance" "nombre" {  
  ami           = var.ami  
  instance_type = var.instance_type  
  availability_zone = var.az  
  key_name      = var.key_name  
  vpc_security_group_ids = var.security_group  
  tags = {Name = var.nombre_instancia}  
  ebs_block_device {  
    device_name = "/dev/sda1"  
    volume_size = var.ebs_volume_size  
  }  
}
```

Fuente: elaboración propia.

En el archivo variables.tf, reside la configuración de los valores definidos para cada una de las variables que se utilizan en el archivo main.tf.

Figura 6. **Plantilla de archivo variables.tf**

```
variable "nombre_variable" {  
  type = tipo_dato  
  default = valor  
}
```

Fuente: elaboración propia.

Tabla VI. **Explicación de variables**

<b>Variable</b>	<b>Tipo</b>	<b>Descripción</b>
key_name	string	Llave SSH creada anteriormente en la cuenta de AWS.
security_group	string	Grupo de seguridad creado en la cuenta de AWS.
name	string	Si no se asigna un nombre específico, se asigna uno por defecto.
ami	string	Id de la imagen que las instancias usarán.
az	string	Región de disponibilidad que alojará las instancias.
instance_type	string	Recursos de RAM y CPU.
ebs_volume_size	number	Si no se asigna una cantidad de GB específico, se asigna 8GB por defecto.

Fuente: elaboración propia.

Con el archivo output.tf, se imprime en consola datos de las instancias al momento de completar el flujo de trabajo de Terraform o utilizando el comando terraform output.

Figura 7. **Plantilla de archivo output.tf**

```
output "IP_instancia" {  
  value = aws_instance.nombre_recurso.public_ip  
}
```

Fuente: elaboración propia.

En el archivo provider.tf se define al proveedor. Con la herramienta aws-cli se realiza la configuración de los datos de la cuenta de AWS, por razones de seguridad no es una buena práctica colocar en un archivo las credenciales de un usuario. Por lo que se recomienda crear un usuario IAM con los permisos para manejo de EC2 (AWS, s.f).

Figura 8. **Archivo provider.tf**

```
provider "aws" {  
}
```

Fuente: elaboración propia.

Figura 9. **Configuración de cuenta AWS**

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o "awscliv2.zip"
$ unzip awscliv2.zip
$ sudo ./aws/install

$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

Fuente: elaboración propia.

Figura 10. **Flujo de trabajo de Terraform**

```
$ terraform init
$ terraform plan
$ terraform apply
```

Fuente: elaboración propia.

Se comprueba en la cuenta de AWS, que las instancias hayan sido creadas correctamente, y se continúa a la configuración de Ansible.

### 5.3. **Configuración de instalación de software con Ansible**

Se continúa con el proceso de automatización de la infraestructura ahora con la categoría de administración de configuración de software. Para ello se utiliza Ansible, ya que tiene una baja curva de aprendizaje.

Figura 11. **Instalación de Ansible**

```
$ python3 --version
$ sudo apt -y install sshpass
$ sudo apt update
$ sudo apt install ansible -y
$ ansible --version
```

Fuente: elaboración propia.

Al terminar la instalación de Ansible se procede al directorio `/etc/ansible` donde se encuentran los archivos `ansible.cfg` y `hosts`.

Figura 12. **Archivo hosts**

```
[nombre_grupo_maquinas:vars]
dirección IP maquina trabajadora

[grupo_maquinas:vars]
ansible_user = nombre_usuario de maquina trabajadora
ansible_ssh_private_key_file = direccion local de llave privada
ansible_python_interpreter = /usr/bin/python3
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

Fuente: elaboración propia.

En el archivo `hosts` se crean grupos de máquinas, colocando las direcciones IP de las máquinas trabajadoras correspondientes. En las declaraciones, `grupo_maquinas:vars`, se sitúan las variables que nos servirán para hacer exitosa la conexión a los `hosts` de cada grupo.

Tabla VII. **Variables**

<b>Variable</b>	<b>Descripción</b>
ansible_user	Con Ubuntu el usuario es ubuntu.
ansible_ssh_private_key_file	Dirección local de la llave privada.
ansible_python_interpreter	Se debe especificar Python3, ya que Ansible utiliza Python2 por defecto.
ansible_ssh_common_args	Permite de manera segura hacer un bypass de las nuevas conexiones SSH.

Fuente: elaboración propia.

Para verificar que la conexión es exitosa, se ejecuta un ping a cada uno de los grupos de máquinas configurados en el archivo hosts.

Figura 13. **Verificación de conexión**

```
$ ansible nombre_de_grupo_de_maquinas -m ping
```

Fuente: elaboración propia.

Luego de comprobar la conexión se crean archivos llamados playbooks en la carpeta ansible, donde se declarará lo necesario para cada grupo de hosts trabajadores:

Tabla VIII. Paquetes para instalar en máquina de MySQL

Paquete	Descripción
update and upgrade	Actualiza el ambiente de desarrollo.
MySQL_server	Instala MySQL
sysbench	Instala sysbench.

Fuente: elaboración propia.

Figura 14. Playbook para la máquina de MySQL

```
---
- name: MySQL Configuration
  hosts: mysql
  become: yes
  tasks:
    - name: Update and upgrade
      apt:
        update_cache=yes
        upgrade=yes
    - name: Install MySQL Server
      apt:
        name=mysql-server
    - name: Download and run binary package
      shell: curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.sh | sudo bash
      args:
        warn: no
    - name: Install Sysbench
      apt:
        name=sysbench
    - name: Check Sysbench installation
      command: which sysbench
      register: sysbench_installed
      ignore_errors: True
      changed_when: False
```

Fuente: elaboración propia.

Tabla IX. Paquetes para máquina de MongoDB

Paquete	Descripción
update and upgrade	Actualiza el ambiente de desarrollo.
mongodb	Instala MongoDB.
sysbench	Instala sysbench
libmongoc-dev	Librería de MongoDB para desarrollar aplicaciones de C.
libbson-dev	Genera y analiza gramaticalmente archivos BSON,
luarocks	Instala el administrador para módulos tipo Lua.
mongorovert	Permite ejecutar sysbench con MongoDB.

Fuente: elaboración propia.

Figura 15. Playbook para la maquina con MongoDB

```

---
- name: MongoDB Configuration
  hosts: mongodb
  become: yes
  tasks:
    - name: Update and upgrade
      apt:
        update_cache=yes
        upgrade=yes
    - name: Install MongoDB Server
      apt:
        name=mongodb
    - name: Download and run binary package
      shell: curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.sh | sudo bash
      warn: no
    - name: Install Sysbench
      apt:
        name=sysbench
    - name: Check Sysbench installation
      command: which sysbench
      register: sysbench_installed
      ignore_errors: True
      changed_when: False
    - name: Install Drivers
      apt:
        pkg:
          - libmongoc-dev
          - libbson-dev
          - luarocks
    - name: Install Mongorovert driver
      command: luarocks install mongorovert

```

Fuente: elaboración propia.

Tabla X. Paquetes para instalar en máquina principal de TiDB

Paquete	Descripción
update and upgrade	Actualiza el ambiente de desarrollo.
MySQL_server	Instala MySQL
sysbench	Instala sysbench
sshpas	Instala sshpas

Fuente: elaboración propia.

Figura 16. Playbook para la máquina de TiDB

```

---
- name: TiDB Configuration
  hosts: tidb
  become: yes
  tasks:
    - name: Update and upgrade
      apt:
        update_cache=yes
        upgrade=yes
    - name: Install MySQL Server
      apt:
        name=mysql-server
    - name: Download and run binary package
      shell: curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.sh | sudo bash
      args:
        warn: no
    - name: Install Sysbench
      apt:
        name=sysbench
    - name: Check Sysbench installation
      command: which sysbench
      register: sysbench_installed
      ignore_errors: True
      changed_when: False
    - name: Install SSHPASS
      apt:
        name=sshpas
  
```

Fuente: elaboración propia.

Tabla XI. Paquetes para instalar en máquinas requeridas por TiDB

Paquete	Descripción
update and upgrade	Actualiza el ambiente de desarrollo.
sshpas	Instala sshpas

Fuente: elaboración propia.

Figura 17. **Playbook para máquinas requeridas por TiDB**

```

---
- name: TiDB Configuration
  hosts: tidb
  become: yes
  tasks:
    - name: Update and upgrade
      apt:
        update_cache=yes
        upgrade=yes
    - name: Install MySQL Server
      apt:
        name=mysql-server
    - name: Download and run binary package
      shell: curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.sh | sudo bash
      args:
        warn: no
    - name: Install Sysbench
      apt:
        name=sysbench
    - name: Check Sysbench installation
      command: which sysbench
      register: sysbench_installed
      ignore_errors: True
      changed_when: False
    - name: Install SSHPASS
      apt:
        name=sshpas
  
```

Fuente: elaboración propia.

Para poder instalar los paquetes necesarios se debe de utilizar las opciones y módulos de Ansible, a continuación, un resumen de los recursos que se utilizaron en los playbooks.

Tabla XII. **Módulos y opciones usados en playbooks**

Declaración	Descripción
name	Identifica el flujo, tarea o un paquete.
hosts	Identifica el grupo de máquinas.
become	Convierte en usuario root.
tasks	Identifica las tareas que se realizarán.
apt	Instala paquetes apt.
shell	Ejecuta comandos con bin/sh.
command	Ejecuta comandos con bin/bash.
register	Retorna texto al concluir la acción previa.
ignore_errors	Ignora errores al ejecutar una tarea.
changed_when	Indica un cambio en una tarea.
pkg	Instala una lista de paquetes con apt.

Fuente: elaboración propia.

Por último, se ejecuta en la terminal el siguiente comando para hacer efectiva las configuraciones declaradas.

Figura 18. **Ejecución de playbooks**

```
$ sudo ansible-playbook nombre_de_playbook
```

Fuente: elaboración propia.

## 6. IMPLEMENTACIÓN DE EXPERIMENTO

### 6.1. Implementación

Al tener configurada la arquitectura, se procederá con la configuración e implementación del experimento utilizando las maquinas creadas para la administración de tanto MySQL, MongoDB y TiDB. Se realizará 3 veces cada experimento de cada base de datos para calcular un promedio.

### 6.2. Configuración de MySQL para implementación de experimento

- Crear un usuario nuevo en la dirección localhost e identificarlo con una contraseña.
- Otorgarle a este nuevo usuario todos los privilegios.
- Hacer efectivo estos privilegios.
- Crear una nueva base de datos.

Figura 19. Configuración de base de datos MySQL

```
$ sudo mysql
mysql> create user 'usuario_nuevo'@'localhost' identified by 'contraseña_nueva';
mysql> grant all privileges on *.* to 'usuario_nuevo'@'localhost' with grant option;
mysql> flush privileges;
mysql> create database nueva_base_datos;
```

Fuente: elaboración propia.

### 6.2.1. Preparación de experimento de MySQL

Se usarán los parámetros definidos anteriormente directamente en el comando y otros en un archivo config.

Figura 20. Contenido archivo config

```
mysql-host=localhost
mysql-port=3306
mysql-user=usuario_creado
mysql-password=contraseña_creada
mysql-db=base_de_datos_creada
time=120
threads=128
report-interval=10
db-driver=mysql
```

Fuente: elaboración propia.

Figura 21. Preparación de experimento de MySQL

```
$ sysbench --config-file=config /usr/share/sysbench/oltp_insert.lua --tables=15 --table-size=500000 prepare
```

Fuente: elaboración propia.

### 6.2.2. Ejecución de experimento de MySQL

Los resultados se guardarán en archivos secuenciales MySQL\_test.log.

Figura 22. Ejecución de experimento de MySQL

```
$ sysbench --config-file=config /usr/share/sysbench/oltp_insert.lua --tables=15 --table-size=500000 run >> mysql_test.log
```

Fuente: elaboración propia.

### 6.3. Configuración de MongoDB para implementación de experimento

- Crear una nueva base de datos.

Figura 23. Configuración de base de datos MongoDB

```
$ mongo
mongo> use mongodb_test
```

Fuente: elaboración propia.

#### 6.3.1. Preparación de experimento de MongoDB

- Se utilizan los parámetros definidos anteriormente directamente en el comando y otros en un config.
- Se debe de crear el archivo `oltp_mongo.lua`, en el directorio `/usr/share/sysbench/`, el contenido de este archivo se encuentra en el repositorio `sysbench-mongodb-lua` (Stroganov, s.f). Este archivo es una modificación de la prueba de `oltp_insert` para ejecutarlo con el driver de `mongodb`.

Figura 24. Contenido archivo config

```
mongodb-db=mongodb_test
mongodb-host=localhost
mongodb-port=27017
rand-type=pareto
threads=128
time=120
```

Fuente: elaboración propia.

## Figura 25. Preparación de experimento de MongoDB

```
$ sysbench --config-file=config /usr/share/sysbench/oltp_mongo.lua --tables=15 --table-size=500000 prepare
```

Fuente: elaboración propia.

### 6.3.2. Ejecución de experimento de MongoDB

Los resultados se guardarán en archivos secuenciales mongodb\_test.log.

## Figura 26. Ejecución de experimento de MongoDB

```
$ sysbench --config-file=config /usr/share/sysbench/oltp_insert.lua --tables=15 --table-size=500000 run >> mongodb_test.log
```

Fuente: elaboración propia.

## 6.4. Configuración de TiDB para implementación de experimento

- Instalar TiUP,
- Instalar el componente clúster.
- Inicializar el archivo que maneja la topología del clúster.
- Ejecutar el comando de despliegue del clúster.
- Verificar el estado del clúster.
- Iniciar el clúster.
- Crear una base de datos desde la maquina administradora de TiDB en la instancia del clúster llamada TiDB\_server.

### 6.4.1. Instalación de la herramienta TiUP

- Se utiliza curl para descargar el archivo install.sh e instalar TiUP.
- Se actualizan las variables de entorno para acceder al comando tiup.
- Por último, se verifica la instalación de tiup con el comando which.

Figura 27. Instalación de la herramienta TiUP

```
$ curl --proto '=https' --tlsv1.2 -sSf https://tiup-mirrors.pingcap.com/install.sh | sh
$ source /home/ubuntu/.bashrc
$ which tiup
```

Fuente: elaboración propia.

### 6.4.2. Instalar el componente clúster

Se procede a instalar el componente clúster para poder manejar los diferentes atributos que se requerirán para poder correr TiDB.

Figura 28. Instalación del componente clúster

```
$ tiup cluster
$ tiup update -self && tiup update cluster
```

Fuente: elaboración propia.

### 6.4.3. Inicializar el archivo que maneja el clúster

Se usará una plantilla de un clúster en el archivo cluster.yaml. Se colocarán las direcciones IP y los nombres de cada pod.

Figura 29. **Inicializar el archivo cluster.yaml**

```
$ tiup cluster template --local > cluster.yaml
```

Fuente: elaboración propia.

Figura 30. **Contenido del archivo cluster.yaml**

```
pd_servers:
- host: direccion_IP

tidb_servers:
- host: direccion_IP

tikv_servers:
- host: direccion_IP

monitoring_servers:
- host: direccion_IP

grafana_servers:
- host: direccion_IP
```

Fuente: elaboración propia.

#### 6.4.4. **Despliegue del clúster**

- Se verifican potenciales errores de forma automática con los comandos check y apply.
- Si no existe ningún error se procede a desplegar el clúster con el comando deploy. Son necesarias las llaves de acceso SSH de la máquina administradora de TiDB para realizar los comandos.

Figura 31. **Despliegue del clúster**

```
$ tiup cluster check ./cluster.yaml --user ubuntu -i ./llave.pem
$ tiup cluster check ./cluster.yaml --apply --user ubuntu -i ./llave.pem
$ tiup cluster deploy nombre_cluster v6.0.0 ./cluster.yaml --user ubuntu -i ./llave.pem
```

Fuente: elaboración propia.

#### 6.4.5. Verificar el estado del clúster

Se deben de listar los clústeres creados y verificar el estado, utilizando el nombre que se le dio en el comando anterior.

Figura 32. Verificar el estado del clúster

```
$ tiup cluster list  
$ tiup cluster display nombre_de_cluster
```

Fuente: elaboración propia.

#### 6.4.6. Iniciar el clúster

Se inicializa el clúster para correr los diferentes servicios necesarios para que TiDB pueda ejecutarse de forma exitosa.

Figura 33. Iniciar el clúster

```
$ tiup cluster start nombre_de_cluster
```

Fuente: elaboración propia.

#### 6.4.7. Creación de base de datos

Después de haber creado el clúster de TiDB se crea la base de datos usando la terminal MySQL y se conecta al host de TiDB\_server.

Figura 34. **Creación de base de datos**

```
$ mysql -u root -h direccion_IP_tidb_server_host -P 4000
mysql> set global tidb_disable_txn_auto_retry = off;
mysql> create database nombre_base_datos;
```

Fuente: elaboración propia.

#### 6.4.8. Preparación de experimento para la base de datos TiDB

- Se utilizan los parámetros definidos anteriormente directamente en el comando y otros en el archivo config.
- Se debe sobrescribir el archivo `oltp_common.lua`, en el directorio `/usr/share/sysbench/`, con el contenido de un parche (Jacky, s.f).

Figura 35. **Contenido archivo config**

```
mongodb-db=mongodb_test
mongodb-host=localhost
mongodb-port=27017
rand-type=pareto
threads=128
time=120
```

Fuente: elaboración propia.

Figura 36. **Preparación de experimento de TiDB**

```
$ sysbench --config-file=config oltp_point_select --tables=8 --table-size=500000 prepare
```

Fuente: elaboración propia.

#### 6.4.9. Ejecución de experimento con la base de datos TiDB

Se ejecutará el experimento 3 veces para contar con diferentes valores al hacer el análisis. Los resultados se guardarán en archivos secuenciales llamados TiDB\_test.log.

Figura 37. Ejecución de experimento de TiDB

```
$ sysbench --config-file=config oltp_point_select --tables=8 --table-size=500000 run >> tidb_test.log
```

Fuente: elaboración propia.



## 7. ANÁLISIS DE RESULTADOS

Al culminar los experimentos, se analizarán los resultados a partir de los indicadores propuestos en los capítulos anteriores con ayuda de instrumentos de estadística descriptiva.

### 7.1. Resultado de MySQL

En las siguientes tablas se presentan de forma correspondiente los resultados y el promedio del experimento de MySQL.

Tabla XIII. **Resultados de experimento de MySQL**

<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total Seg</b>	<b>Número total de eventos</b>
1	16.37	600.0500	4691530
2	19.02	600.0356	4036146
3	19.68	600.0474	3900689

Fuente: elaboración propia.

Tabla XIV. **Promedio de datos de experimento de MySQL**

<b>Promedio</b>			
<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total seg</b>	<b>Número total de eventos</b>
1	18.36	600.0443	4209455

Fuente: elaboración propia.

## 7.2. Resultado de MongoDB

En las siguientes tablas se presentan de forma correspondiente los resultados y el promedio del experimento de MongoDB.

Tabla XV. **Resultados de experimento de MongoDB**

<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total seg</b>	<b>Número total de eventos</b>
1	449.29	600.3254	170974
2	451.37	600.3219	170200
3	451.79	600.3018	170029

Fuente: elaboración propia.

Tabla XVI. **Promedio de datos de experimento MongoDB**

<b>Promedio</b>			
<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total seg</b>	<b>Número total de eventos</b>
1	450.82	600.3164	170401

Fuente: elaboración propia.

## 7.3. Resultado de TiDB

En las siguientes tablas se presentan de forma correspondiente los resultados y el promedio del experimento de MySQL.

Tabla XVII. **Resultados de experimento de TiDB**

<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total seg</b>	<b>Número total de eventos</b>
1	16.03	600.0297	4791790
2	15.95	600.0312	4813582
3	16.07	600.0395	4779241

Fuente: elaboración propia.

Tabla XVIII. **Promedio de datos de experimento de TiDB**

<b>Promedio</b>			
<b>N</b>	<b>Latencia media mseg</b>	<b>Tiempo Total seg</b>	<b>Número total de eventos</b>
1	16.02	600.0335	4794871

Fuente: elaboración propia.

#### **7.4. Análisis de resultados**

Con los datos de los experimentos realizados se observa que TiDB tuvo un mejor rendimiento a nivel de tiempo y latencia. Si tomamos en cuenta que se usó una carga de datos que se podría considerar Big Data, se concluye que bajo los mismos parámetros y las mismas características de software y hardware la base de datos que tiene un mejor manejo relacionado a la Big Data es TiDB.

Se debe tomar en cuenta que, aunque en la teoría y bajo un experimento controlado puede considerarse la mejor opción, para una organización pequeña o mediana puede convertirse en un alto costo financiero. Aunque TiDB ofrece una curva de aprendizaje baja, el manejo de una base de datos distribuida puede convertirse en un problema para un equipo que no tiene la experiencia manejando estos conceptos.

Por lo que, aunque poner en práctica un nuevo concepto es algo que siempre se debe de impulsar desde cualquier organización para mantenerse al tanto de la tecnología, siempre hay que analizar cada proyecto antes de elegir una opción de tecnología para obtener un mejor resultado.

## CONCLUSIONES

1. La mejor opción para manejar datos de Big Data en nuestro experimento es TiDB, aunque debe analizarse lo que necesita un proyecto en específico antes de tomar la decisión de que tecnología utilizar.
2. El estado del arte en relación con NewSQL es un tema que está en constante evolución.
3. Los casos de éxito donde se puede utilizar NewSQL puede mejorar la velocidad de eventos de consultas y de transacción ya que utiliza un modelo distribuido.
4. Con NewSQL se cumplen las 3 partes del teorema de CAP.



## RECOMENDACIONES

1. Conceptualizar el teorema de CAP para tener un entendimiento integral de bases de datos.
2. Profundizar en la base de datos que más interés haya despertado a la hora de leer este documento y encontrar otras formas para probarlo o utilizarlo en un proyecto.
3. Realizar el experimento propuesto para aprender a utilizar diferentes herramientas de automatización como un conocimiento complementario.



## REFERENCIAS

1. Barzu, C. (2017). *Estudio del rendimiento de sistemas de gestión de bases de datos New SQL* (tesis de maestría). Universidad Politécnica de Madrid, España. Recuperado de <https://oa.upm.es/47291/>.
2. Bases de datos SQL | AWS. (2021). AWS. Recuperado de <https://aws.amazon.com/es/relational-database/>.
3. Creating an IAM user in your AWS account - AWS Identity and Access Management. (2021). AWS. Recuperado de [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_users\\_create.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html).
4. Cupas, C. (25 de octubre, 2021). *Qué es el Teorema CAP y cómo afecta al elegir la BBDD*. OpenWebinars. Recuperado de <https://openwebinars.net/blog/que-es-el-teorema-cap-y-como-afecta-al-elegir-la-base-de-datos/>.
5. Explicación Sobre Las Bases De Datos NoSQL. (2021). *MongoDB*. Recuperado de <https://www.mongodb.com/es/NoSQL-explained>.
6. Goldberg, S. (11 de agosto, 2020). *Deep Dive: NewSQL Databases*. *DZone*. Recuperado de <https://dzone.com/articles/deep-dive-NewSQL-databases>.

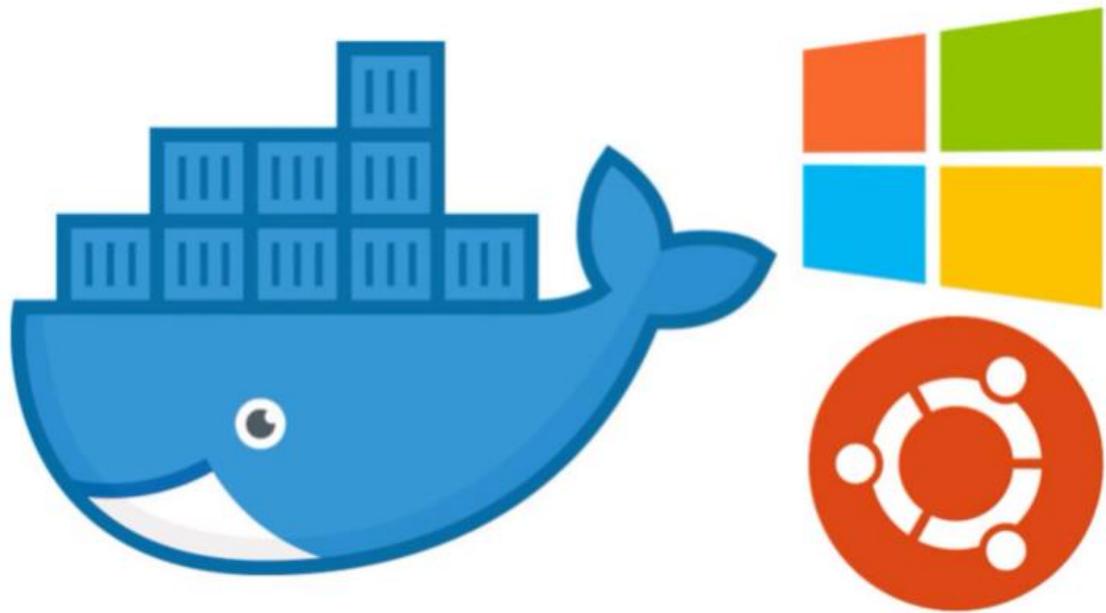
7. Huang, E. (2020). *TiDB 3.0. PingCAP*. Recuperado de [https://docs.google.com/presentation/d/1tJQO2TR\\_K33VnpMvALHkRMt8k8DSKml5ldPJYd8gV4/edit#slide=id.g512b39c455\\_1\\_7](https://docs.google.com/presentation/d/1tJQO2TR_K33VnpMvALHkRMt8k8DSKml5ldPJYd8gV4/edit#slide=id.g512b39c455_1_7).
8. Jacky, S. (2019). *pingcap/tidb-bench*. *GitHub*. Recuperado de <https://github.com/akopytov/sysbench>.
9. Johansson, M. y Jonatan, R. (2020). *Performance comparison between NewSQL and SQL: Sharded TiDB vs MariaDB* (tesis de licenciatura). Universidad de Skövde, Suecia. Recuperado de <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1449822&dswid=1708>.
10. Kaur, Karambir, and Monika Sachdeva. (2017). *Performance Evaluation of NewSQL Databases*. India: International Conference on Inventive Systems and Control. Recuperado de <https://ieeexplore.ieee.org/abstract/document/8068585>.
11. Khasawneh, T. Mahmoud A. y Ali S. (Abril, 2020). *SQL, NewSQL, and NOSQL Databases: A Comparative Survey*. Jordania: IEEE. Recuperado de <https://ieeexplore.ieee.org/abstract/document/9078970>.
12. Kopytov, A. (2021). *akopytov/sysbench*. *GitHub*. Recuperado de <https://github.com/akopytov/sysbench>.
13. La plataforma de alojamiento MySQL mejor administrada para su aplicación. (2021). *Geekflare*. Recuperado de <https://geekflare.com/es/MySQL-hosting-platform/>.

14. Liu, K. (20 de agosto, 2020). *TiDB Cloud: Managed SQL at Scale on AWS and GCP*. *PingCAP*. Recuperado de <https://en.pingcap.com/blog/TiDB-cloud-managed-sql-at-scale-on-aws-and-gcp/>.
15. Marr, B. (21 de mayo, 2018). *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read*. *Forbes*. Recuperado de <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=5cc5e29260ba>.
16. MongoDB Cloud. (2021). *MongoDB*. Recuperado de <https://www.mongodb.com/cloud>.
17. MySQL Customers. (2022). *MySQL*. Recuperado de <https://www.mysql.com/customers/>.
18. OLTP, OLAP, & HTAP. (2022). *Hydra*. Recuperado de <https://docs.hydra.so/concepts/oltp-olap-and-htap>.
19. Our Customers. (2022). *MongoDB*. Recuperado de <https://www.mongodb.com/who-uses-mongodb>.
20. Petroc, T. (10 de agosto, 2022). *Most popular database management systems 2022*. *statista*. Recuperado de <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>.

21. Raj, P. y Ganesh C. (2018). *A Deep Dive into NoSQL Databases: The Use Cases and Applications*. India: Elsevier Science. Recuperado de <https://doi.org/10.1016/bs.adcom.2018.01.002>.
22. Sosa, W. (2019). *Big data: Breve manual para conocer la ciencia de datos que ya invadió nuestras vidas*. Buenos Aires. Siglo XXI Editores. doi: 978-987-629-926-8.
23. Stroganov, A. (2017). *oltp-mongo.lua*. *GitHub*. Recuperado de <https://github.com/Percona-Lab/sysbench-mongodb-lua/blob/master/oltp-mongo.lua>.
24. Vileikis, L. (28 de enero de 2021). *Big Data + MySQL = Mission Impossible*. *Arctype*. Recuperado de <https://arctype.com/blog/MySQL-storage-engine-big-data/>.
25. Wang, C. (9 de septiembre, 2015). *Introducing mongorover, a New Experimental MongoDB Driver for Lua*. *MongoDB*. Recuperado de <https://www.mongodb.com/blog/post/introducing-mongorover-a-new-experimental-mongodb-driver-for-lua>.
26. What Is Big Data. (2022). *MongoDB*. Recuperado de <https://www.mongodb.com/basics/big-data-explained>.
27. What is Online Transaction Processing (OLTP). (2021). *Oracle*. Recuperado de <https://www.oracle.com/database/what-is-oltp/>.

## APÉNDICE

### Apendice 1. Instalar WSL2 en Windows Home



Fuente: elaboración propia, empleando captura de pantalla,  
<https://www.youtube.com/watch?v=AvdsNOgiiRk>.



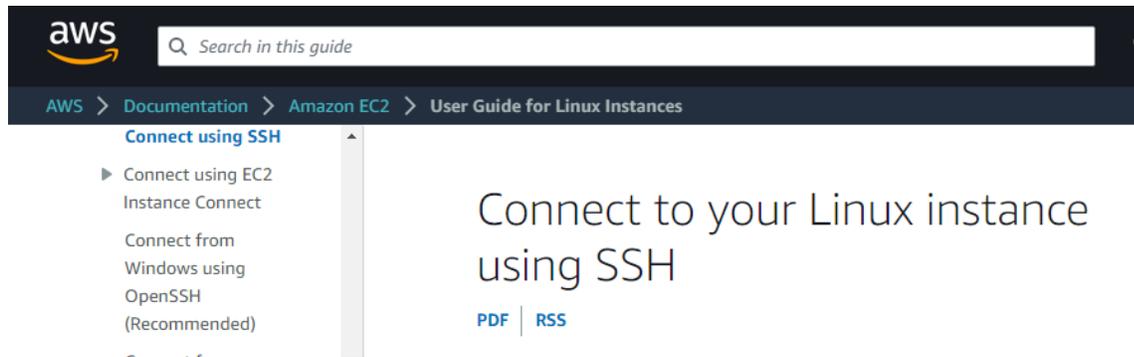
## ANEXOS

### Anexo 1. **Como usar TiDB en una EC2 de AWS**



Fuente: PingCAP (2021). *What is TiDB*. Consultado el 27 de abril de 2022. Recuperado de <https://www.youtube.com/watch?v=R7F7vPnaoTY>.

## Anexo 2. Conectar EC2 de AWS a cliente SSH



Fuente: AWS (2022). *DevOps*. Consultado el 27 de abril de 2022. Recuperado de <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>.