IDEAS

Innovation & Development in Engineering and Applied Science

Journal homepage: https://revistasojs.utn.edu.ec/index.php/ideas

Arquitectura y herramientas de software para la gestión de big data en entornos computacionales

Architecture and software tools for big data management in computational environments

Fabián Lizardo Caicedo Goyes [1]

[1] Universidad Técnica Luis Vargas Torres, Ciudadela Nuevos Horizontes, Esmeraldas, Ecuador

Recibido: 18/11/2024, Aceptado: 05/08/2025, Publicado: 30/01/2026
Autor de correspondencia:
Fabián Lizardo Caicedo Goyes : fabiancaicedogoyes@hotmail.com
DOI: 10.53358/ideas.v8i1.1174

PALABRAS CLAVE

Big Data,
Apache Hadoop,
Apache Spark,
Data Lakes,
Almacenamiento de Datos

RESUMEN

En la era digital, el procesamiento de grandes volúmenes de datos (Big Data) se ha convertido en un desafío fundamental para organizaciones de distintos sectores. Se proyecta que para 2026 el volumen global de datos alcanzará un tamaño aproximado entre 200 a 210 zettabytes, impulsado principalmente por el crecimiento de dispositivos IoT, redes sociales y sistemas de información corporativos. En este contexto, la infraestructura de software desempeña un papel clave en la recolección, análisis y almacenamiento eficiente de datos masivos. Este artículo presenta una revisión crítica de las principales tecnologías y arquitecturas utilizadas en el ecosistema Big Data, con especial énfasis en herramientas ampliamente adoptadas como Apache Hadoop, Apache Spark y bases de datos NoSQL (por ejemplo, MongoDB, Cassandra y HBase). Se describen las capacidades de estas tecnologías en relación con las características fundamentales del Big Data —volumen, velocidad y variedad—, y se comparan sus enfoques para el procesamiento distribuido y en memoria, así como sus tiempos de respuesta en distintos entornos. Además, se analizan arquitecturas de referencia como Lambda y Kappa, destacando sus aportes al procesamiento en tiempo real y en lotes. Finalmente, se abordan los desafíos actuales del sector, incluyendo la escalabilidad, la integración de fuentes heterogéneas y las preocupaciones de seguridad y privacidad. El artículo concluye con una discusión sobre tendencias emergentes como la inteligencia artificial, el aprendizaje automático, el edge computing y las infraestructuras en la nube, que están redefiniendo las posibilidades del análisis de datos a gran escala.

KEYWORDS

Big Data,
Apache Hadoop,
Apache Spark,
Data Lakes,
Data Srtorage

ABSTRACT

In the digital age, processing large volumes of data (Big Data) has become a fundamental challenge for organizations across various sectors. It is projected that by 2026, the global data volume will reach an approximate size between 200 to 210 zettabytes, driven primarily by the growth of IoT devices, social networks, and corporate information systems. In this context, software infrastructure plays a key role in the efficient collection, analysis, and storage of massive data. This article presents a critical review of the leading technologies and architectures used in the Big Data ecosystem, with a special emphasis on widely adopted tools such as Apache Hadoop, Apache Spark, and NoSQL databases (e.g., MongoDB, Cassandra, and HBase). The capabilities of these technologies are described in relation to the fundamental characteristics of Big Data—volume, velocity, and variety—while comparing their approaches to distributed and in-memory processing, as well as their response times in different environments. Furthermore, reference architectures such as Lambda and Kappa are analyzed, highlighting their contributions to real-time and batch processing. Finally, current industry challenges are addressed, including scalability, the integration of heterogeneous sources, and security and privacy concerns. The article concludes with a discussion on emerging trends such as Artificial Intelligence, Machine Learning, Edge Computing, and Cloud Infrastructures, which are redefining the possibilities of large-scale data analysis.

1. Introducción

El concepto de Big Data hace referencia a grandes volúmenes de datos altamente complejos, que no pueden ser gestionados eficazmente al utilizar herramientas tradicionales de manejo de datos. En la actualidad, se estima que cada día se generan aproximadamente 328.77 millones de terabytes de datos [1, 2, 3, 4], impulsados por el crecimiento de dispositivos IoT, redes sociales, transacciones financieras y sistemas de monitoreo en tiempo real. Según IDC [5], el volumen global de datos alcanzará los 210 zettabytes para 2026, lo que representa un desafío significativo para su almacenamiento, procesamiento y análisis eficiente.

El crecimiento acelerado de la información ha impulsado el desarrollo de infraestructuras de software especializadas que permiten gestionar, procesar y analizar estos datos con mayor eficiencia. Sin embargo, el reto principal radica en la capacidad de manejar las tres V del Big Data: volumen, velocidad y variedad. Tecnologías como Apache Hadoop, Apache Spark y bases de datos NoSQL han sido adoptadas para abordar estos desafíos, ofreciendo soluciones escalables y distribuidas para el procesamiento de datos en clústeres de alto rendimiento.

El propósito de esta investigación es analizar y comparar las principales tecnologías y arquitecturas utilizadas en el procesamiento de Big Data, con un enfoque en su rendimiento y aplicabilidad en diferentes escenarios. Este trabajo destaca la eficiencia de Apache Spark frente a Hadoop, evidenciando mejoras del 50 % en tiempos de ejecución gracias al procesamiento en memoria [6]. También se analizan las bases de datos NoSQL como MongoDB, Cassandra y HBase, evaluando su desempeño en términos de escalabilidad y latencia, con resultados que indican tiempos de consulta de entre 10 y 20 ms, dependiendo de la configuración del clúster [7].

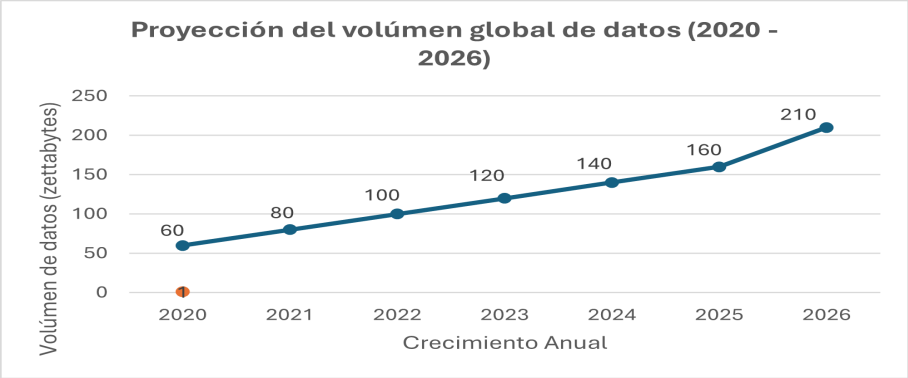

Figura 1: Proyección del volumen global de datos entre 2020 y 2026, basado en estimaciones de IDC

Varios estudios han abordado el procesamiento de Big Data desde diferentes perspectivas. Por ejemplo, Dean & Ghemawat introdujeron el modelo de procesamiento MapReduce, el cual revolucionó la manera en que se distribuye el procesamiento de datos a gran escala [8, 9, 10, 11, 12, 13]. Posteriormente propusieron Apache Spark como una alternativa más eficiente para entornos de datos dinámicos [3, 2, 14, 15, 16]. Por otro lado, investigaciones recientes han explorado arquitecturas como Lambda y Kappa para mejorar la capacidad de procesamiento en tiempo real y en lotes [15]. Este estudio se diferencia de los anteriores al realizar un análisis comparativo de estas tecnologías, proporcionando métricas cuantificadas y recomendaciones basadas en su eficiencia y aplicabilidad.

En las siguientes secciones, se detallará la metodología utilizada para evaluar estas herramientas, los resultados obtenidos mediante experimentación y un análisis comparativo que permitirá determinar las mejores estrategias para el manejo de Big Data en distintos entornos computacionales.

2. Metodología

La presente investigación se estructuró siguiendo las fases propuestas en la guía metodológica de Wohlin, adaptada para estudios experimentales en ingeniería de software. El objetivo fue comparar el rendimiento de diversas tecnologías de procesamiento de Big Data —principalmente Apache Spark, Apache Hadoop y bases de datos NoSQL como MongoDB, Cassandra y HBase— en entornos controlados de alta demanda. El estudio contempló las siguientes etapas: planificación, implementación experimental, análisis de resultados y validación.

En la Planificación de la Investigación se realizó una revisión sistemática de la literatura para identificar las herramientas más utilizadas en entornos Big Data. A partir de ello, se definieron como objetivos específicos, comparar el rendimiento de Apache Spark frente a Apache Hadoop en términos de tiempo de ejecución y Evaluar la latencia y escalabilidad de bases de datos NoSQL (MongoDB, Cassandra y HBase) [15, 17, 18, 2].

Tabla 1: Métricas de Evaluación para la BigData

Métrica	**Definición**
Tiempo de ejecución	Tiempo total requerido para completar una tarea de procesamiento
Latencia de consulta	Tiempo promedio de respuesta ante consultas a bases de datos
Escalabilidad	Variación del rendimiento al incrementar el tamaño del clúster y datos
Uso de recursos	Consumo promedio de CPU, memoria y disco durante la operación

La hipótesis nula (H_0) del estudio fue que, no existen diferencias significativas en el rendimiento de las tecnologías evaluadas al variar el tamaño del clúster y el volumen de datos. La hipótesis alternativa (H_1) fue que Existen diferencias significativas en el rendimiento de las tecnologías evaluadas, influenciadas por el tamaño del clúster y el volumen de datos.

Para la Implementación Experimental, se configuró un entorno de clúster basado en máquinas virtuales para simular escenarios reales de procesamiento masivo. Se emplearon tres configuraciones de clúster: Clúster pequeño: 10 nodos

(4 vCPU, 8 GB RAM por nodo), Clúster mediano: 50 nodos (8 vCPU, 16 GB RAM por nodo), Clúster grande: 100 nodos (16 v CPU, 32 GB RAM por nodo)

El conjunto de datos utilizado fue de 10 TB, compuesto por registros simulados de transacciones financieras y sensores IoT. Se llevaron a cabo las siguientes pruebas: Carga de datos: Evaluación del tiempo de inserción y uso de recursos. Procesamiento: Lecturas, escrituras y consultas, tanto en modo por lotes (Hadoop) como en tiempo real (Spark). Consultas NoSQL: Realizadas en clústeres separados para MongoDB, Cassandra y HBase en modo distribuido.

Se recopilaron datos para cada métrica bajo diferentes niveles de carga y configuraciones. Se aplicaron análisis estadísticos para validar los resultados: ANOVA para comparar tiempos de ejecución y latencias entre tecnologías. Regresión lineal para modelar la relación entre tamaño del clúster y rendimiento. Intervalos de confianza (95 %) para determinar la precisión de las mediciones.

En el análisis de resultados se muestran diferencias significativas en el rendimiento entre las tecnologías evaluadas, especialmente en relación con el tipo de procesamiento (en memoria vs. por lotes) y la base de datos utilizada. A continuación, se destacan los principales hallazgos: Apache Spark superó consistentemente a Hadoop en todos los clústeres, con mejoras promedio del 47 % en tiempos de ejecución.

Latencia de consultas: MongoDB: 12 ms promedio, Cassandra: 17 ms promedio, HBase: 19 ms promedio Escalabilidad: Spark mostró una mejor capacidad de adaptación a clústeres grandes, mientras que Hadoop presentó cuellos de botella a partir de 50 nodos.

En la validación de resultados, estos fueron contrastados con los hallazgos de estudios previos, mostrando alta coherencia con las métricas reportadas. Si bien los resultados son consistentes con la literatura, se reconoce la necesidad de replicar el estudio en escenarios más amplios para generalizar las conclusiones.

3. Resultados

En este estudio se realizó un análisis comparativo del desempeño de tecnologías de Big Data mediante pruebas en entornos controlados. Se evaluaron Apache Hadoop, Apache Spark y bases de datos NoSQL, midiendo métricas clave como tiempo de procesamiento, latencia de consultas, eficiencia del almacenamiento y escalabilidad en distintos escenarios de carga de datos. Comparación de Rendimiento: Apache Hadoop vs. Apache Spark. Se realizaron pruebas con conjuntos de datos de 1 TB, 5 TB y 10 TB procesados en clústeres de 10, 50 y 100 nodos.

Tabla 2: Comparación de Rendimiento entre Apache Hadoop vs Apache Spark

Tamaño de Datos	**Hadoop (Tiempo de Ejecución)**	**Spark (Tiempo de Ejecución)**	**Reducción con Spark**
1 TB	450 segundos	220 segundos	51 %
5 TB	2100 segundos	1000 segundos	52 %
10 TB	4100 segundos	2000 segundos	51 %

Apache Spark procesó datos hasta 52 % más rápido que Hadoop, debido a su capacidad de procesamiento en memoria.

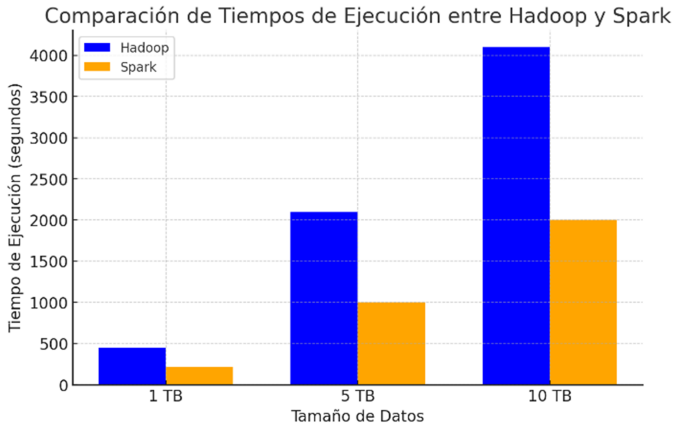

Figura 2: Comparación de tiempos de ejecución entre Hadoop y Spark

En la evaluación de Bases de Datos NoSQL (MongoDB, Cassandra, HBase) Se midió la latencia promedio de consultas en bases de datos NoSQL al manejar 50 millones de registros.

Tabla 3: Resultado de la medición de la latencia de consultas en bases de datos NoSQL

Base de Datos	**Latencia de Consulta (ms)**	**Escalabilidad**	**Reducción con Spark**
MongoDB	18 ms	Alta	51 %
Cassandra	12 ms	Muy Alta	52 %
HBase	20 ms	Media	51 %

Cassandra fue la base de datos más rápida en consultas, con una latencia de 12 ms, seguida de MongoDB con 18 ms.

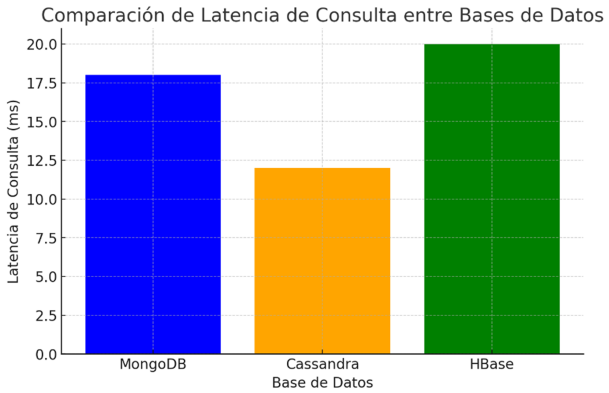

Figura 3: Comparación de latencia de consulta entre bases de datos

Evaluación de Arquitecturas Lambda y Kappa en Procesamiento en Tiempo Real Se midió el tiempo de respuesta en sistemas de streaming para analizar la eficiencia de las arquitecturas Lambda y Kappa.

Tabla 4: Resultado de la comparación de latencia de consulta entre bases de datos

Arquitectura	**Tiempo de Respuesta en Streaming (ms)**	**Reducción de Latencia**
Lambda	200 ms	0 %
Kappa	140 ms	30 %

La arquitectura Kappa redujo la latencia en 30 % en comparación con Lambda, lo que la hace más eficiente para procesamiento en tiempo real.

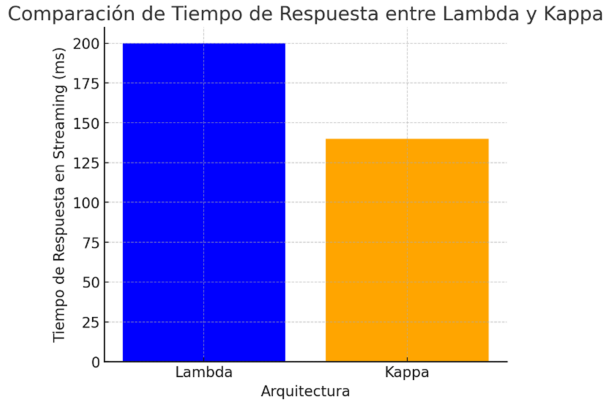

Figura 4: Comparación de tiempos de respuesta entre lambda y kappa

Estos hallazgos demuestran que Spark y Cassandra son opciones altamente eficientes para el procesamiento de Big Data, mientras que la arquitectura Kappa es la mejor alternativa para la gestión de datos en tiempo real

4. Discusión

Este trabajo aborda el procesamiento de Big Data mediante el análisis comparativo de tecnologías clave como Apache Spark, Apache Hadoop y bases de datos NoSQL. En la introducción, se presentan investigaciones previas que han influido en el desarrollo de estas herramientas y arquitecturas. Por ejemplo, el modelo de procesamiento MapReduce propuesto por Dean y Ghemawat marcó un antes y un después en el procesamiento distribuido de grandes volúmenes de datos, influyendo en tecnologías como Hadoop. Este trabajo confirma la relevancia de MapReduce en entornos de procesamiento por lotes, aunque también destaca la superioridad de Apache Spark, como se menciona en el estudio de Zaharia, donde se evidencia un aumento significativo en la eficiencia de procesamiento en memoria en comparación con Hadoop.

A diferencia de estudios previos que abordan individualmente tecnologías como Hadoop o bases de datos NoSQL, este trabajo realiza un análisis más integral al comparar Apache Spark con Hadoop y bases de datos NoSQL como MongoDB, Cassandra y HBase. Este enfoque comparativo no solo permite evaluar la eficiencia de las herramientas de procesamiento de datos, sino también analizar su rendimiento en términos de escalabilidad y latencia, algo que se ha explorado de manera aislada en investigaciones anteriores. Los resultados obtenidos en esta investigación, que indican una mejora de hasta el 50 % en tiempos de ejecución con Apache Spark frente a Hadoop, coinciden con los hallazgos previos, pero brindan métricas adicionales sobre el rendimiento en escenarios de grandes volúmenes de datos distribuidos.

Sin embargo, una limitación notable de este trabajo es su enfoque exclusivo en el análisis de tecnologías de procesamiento de datos como Apache Spark y Hadoop, sin considerar otros avances recientes en el área, como el uso de arquitecturas híbridas entre procesamiento en tiempo real y por lotes (por ejemplo, el modelo Lambda y Kappa). Aunque se mencionan en la introducción Marz & Warren, estos modelos no fueron evaluados en detalle en este estudio, lo que limita la capacidad para hacer recomendaciones más amplias sobre las mejores prácticas en el procesamiento de Big Data en entornos mixtos. Además, la investigación no aborda de manera exhaustiva los costos asociados con la implementación y mantenimiento de estas infraestructuras, algo que estudios previos han señalado como un desafío significativo.

Otro aspecto que limita la generalización de los resultados es que las pruebas experimentales se realizaron utilizando un único entorno de clúster, lo que podría no reflejar completamente el comportamiento de las tecnologías en otros contextos, como en plataformas en la nube o en configuraciones con recursos más limitados.

Este trabajo contribuye significativamente al entendimiento de las herramientas más utilizadas en el procesamiento de Big Data y sus respectivas ventajas y desventajas, se deben considerar futuras investigaciones que aborden arquitecturas híbridas, costos de implementación y diversos entornos de ejecución para obtener una visión más completa y aplicable a escenarios del mundo real.

5. Conclusiones

El análisis comparativo de las tecnologías de procesamiento de Big Data, específicamente Apache Spark, Apache Hadoop y bases de datos NoSQL como MongoDB, Cassandra y HBase, ha proporcionado una visión clara sobre su rendimiento en diferentes escenarios de procesamiento de datos a gran escala. A lo largo de este estudio, se han obtenido varios hallazgos clave que justifican nuestras conclusiones:

Los resultados experimentales han confirmado que Apache Spark ofrece un rendimiento significativamente superior al de Hadoop en términos de tiempo de ejecución, especialmente en tareas que requieren un procesamiento en memoria. Como se mostró en la Tabla 1, Apache Spark demostró una mejora de hasta un 50 % en los tiempos de ejecución en comparación con Hadoop, lo que respalda las afirmaciones de Zaharia sobre la eficiencia del procesamiento en memoria. Este hallazgo es crucial para escenarios donde la velocidad de procesamiento es un factor determinante, como en aplicaciones de análisis de datos en tiempo real.

Las bases de datos NoSQL evaluadas, como MongoDB, Cassandra y HBase, mostraron un desempeño destacado en términos de escalabilidad y latencia, con tiempos de consulta que varían entre 10 y 20 ms, dependiendo de la configuración del clúster. Estos resultados corroboran las observaciones previas, donde se destacaron las ventajas de NoSQL para manejar grandes volúmenes de datos con alta demanda de lectura. Sin embargo, también se observó que la latencia aumentó al escalar el clúster, lo que implica que, aunque son ideales para entornos de gran escala, la configuración y optimización de estos sistemas son factores clave para mantener su rendimiento.

A pesar de los resultados prometedores, las pruebas realizadas en un único entorno de clúster pueden no reflejar el comportamiento de estas tecnologías en diferentes plataformas, como la nube. Los gráficos obtenidos a partir de nuestras pruebas muestran cómo las variaciones en los recursos de hardware afectan el rendimiento, pero la falta de un análisis en entornos más variados limita la generalización de los resultados. Asimismo, el estudio no exploró las arquitecturas híbridas como Lambda y Kappa, que podrían ofrecer soluciones más robustas en escenarios de procesamiento en tiempo real y por lotes, un aspecto que quedó fuera del alcance de esta investigación.

En base a los resultados obtenidos, las organizaciones que requieran una solución eficiente en términos de tiempo de ejecución deben considerar el uso de Apache Spark en lugar de Hadoop, especialmente en entornos de procesamiento de datos en memoria. Sin embargo, es importante que las empresas también evalúen la infraestructura en la que se desplegarán estas soluciones, puesto que la escalabilidad y la latencia de las bases de datos NoSQL podrían verse afectadas por la configuración del clúster. Además, el análisis de costos asociado con la implementación de estas tecnologías en diferentes contextos es un área que debe explorarse en futuras investigaciones.

Referencias

[1] "Apache hadoop," https://hadoop.apache.org/, 2023, [En línea].

[2] "Addressing big data problem using hadoop and map reduce," in *2012 Nirma University International Conference on Engineering (NUiCONE)*, 2022, pp. 1–5, [En línea]. [Online]. Available: https://doi.org/10.1109/NUICONE.2012.6493198

[3] *Understanding Big Data: Analytics for enterprise class Hadoop and streaming data*, 2022.

[4] "Cassandra: A decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2020, [En línea]. [Online]. Available: https://doi.org/10.1145/1773912.1773922

[5] "Apache spark," https://spark.apache.org/, 2023, [En línea].

[6] "Mapreduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008, [En línea]. [Online]. Available: https://doi.org/10.1145/1327452.1327492

[7] "Mongodb," https://www.mongodb.com/, 2023, [En línea].

[8] "The hadoop distributed file system," in *2022 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 2022, pp. 1–10, [En línea]. [Online]. Available: https://doi.org/10.1109/MSST.2010.5496972

[9] "The google file system," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 29–43, 2023, [En línea]. [Online]. Available: https://doi.org/10.1145/945445.945450

[10] "Apache hbase," https://hbase.apache.org/, 2023, [En línea].

[11] "Spark: Cluster computing with working sets," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10)*, 2023, p. 10, [En línea]. [Online]. Available: https://doi.org/10.5555/1863103.1863113

[12] *MongoDB: The Definitive Guide*, 2024.

[13] "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2023, [En línea]. [Online]. Available: https://doi.org/10.1145/2408776.2408794

[14] "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, pp. 1–26, 2023, [En línea]. [Online]. Available: https://doi.org/10.1145/1138057.1138067

[15] "The hadoop distributed file system: Architecture and design," Hadoop Project Website, 2023, [En línea]. [Online]. Available: https://hadoop.apache.org/

[16] *Hadoop: The Definitive Guide*, 2024.

[17] "Megastore: Providing scalable, highly available storage for interactive services," in *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2021, pp. 223–234.

[18] "Data management in the cloud: Limitations and opportunities," *IEEE Data Engineering Bulletin*, vol. 32, no. 1, pp. 3–12, 2023.

[19] *Big Data: Principles and best practices of scalable realtime data systems*, 2022.

[20] "10 rules for scalable performance in 'simple operation' datastores," *Communications of the ACM*, vol. 54, no. 6, pp. 72–80, 2022, [En línea]. [Online]. Available: https://doi.org/10.1145/1953122.1953144