En resumen, **Apache Beam** es un modelo de programación unificado y un conjunto de API de procesamiento de datos de código abierto que simplifica el desarrollo de aplicaciones de procesamiento de datos distribuidos. Con su capacidad para ejecutar código en múltiples motores de ejecución de procesamiento de datos y su conjunto de API expresivas y coherentes, Beam facilita el desarrollo de aplicaciones de procesamiento de datos portátiles y escalables que pueden ejecutarse en una variedad de entornos de procesamiento de dato.

Arquitectura y componentes de Apache Beam

L a **arquitectura** de Apache Beam está diseñada para ofrecer un modelo de programación unificado y portátil que permita el procesamiento de datos distribuidos en una variedad de motores de ejecución de procesamiento de datos. A continuación, describiré los componentes principales de Apache Beam y cómo se interrelacionan en su arquitectura.

Componentes principales

El ***pipeline*** es el componente central de Apache Beam y representa el flujo de datos que será procesado. Un *pipeline* consta de una serie de transformaciones que se aplican a los datos para realizar operaciones como lectura, transformación y escritura.

- **PCollections** (colecciones paralelas). Las PCollections son las estructuras de datos que fluyen a través del *pipeline*. Representan conjuntos de datos distribuidos que pueden ser procesados en paralelo. Las PCollections son inmutables y pueden contener datos en tiempo real o en lotes.

- **Transformaciones.** Las transformaciones son operaciones que se aplican a las PCollections para procesar los datos. Estas transformaciones pueden ser operaciones de lectura, transformación o escritura, y se combinan para formar el grafo de ejecución del *pipeline*.

- ***Runners*** (ejecutores). Los *runners* son los motores de ejecución que ejecutan el *pipeline* y realizan las transformaciones en los datos. Apache Beam proporciona un conjunto de *runners* prediseñados para ejecutar *pipelines* en diferentes entornos, como Apache Flink, Apache Spark, Google Cloud Dataflow y más.

- ***Pipeline options***. Las opciones del *pipeline* son configuraciones que controlan el comportamiento y la ejecución del *pipeline*, como la configuración del entorno de ejecución, la ubicación de los recursos y más.

Proceso de ejecución en Apache Beam

La arquitectura de Apache Beam se basa en un modelo de programación unificado y portátil que permite a los desarrolladores escribir código una vez y ejecutarlo en diferentes motores de ejecución de procesamiento de datos. El **proceso de ejecución** de un *pipeline* en Apache Beam sigue los siguientes pasos:

- **Creación del *pipeline.*** El proceso comienza con la creación del *pipeline*, donde se definen las transformaciones y las fuentes de datos que formarán parte del flujo de datos.

- **Configuración del *pipeline options.*** A continuación, se configuran las opciones del *pipeline*, que incluyen la configuración del entorno de ejecución, las opciones de procesamiento y más.

- **Ejecución del *pipeline.*** Una vez que el *pipeline* está configurado, se ejecuta en un *runner* específico que se encarga de ejecutar las transformaciones en los datos. El *runner* se encarga de gestionar la ejecución del *pipeline*, coordinar las transformaciones y garantizar que los resultados se procesen correctamente.

- **Procesamiento distribuido.** Durante la ejecución del *pipeline*, las transformaciones se aplican de manera distribuida en los datos, lo que permite procesar grandes volúmenes de datos de manera eficiente y escalable.

- **Escritura de resultados.** Finalmente, los resultados del *pipeline* se escriben en un destino específico, como un sistema de almacenamiento o un servicio de análisis, donde pueden ser utilizados para generar informes, tomar decisiones o alimentar otras aplicaciones.

En resumen, la **arquitectura de Apache Beam** está diseñada para ofrecer un modelo de programación unificado y portátil que permite el procesamiento de datos distribuidos en una variedad de motores de ejecución de procesamiento de datos. Con su conjunto de componentes y su arquitectura modular, Apache Beam facilita el desarrollo de aplicaciones de procesamiento de datos escalables y portátiles que pueden ejecutarse en diferentes entornos de procesamiento de datos sin necesidad de modificar el código.

Integraciones de servicios de AWS

Es posible configurar e integrar un destino u origen de datos con un código mínimo. Utiliza las bibliotecas de Amazon Managed Service para Apache Flink para integrarlas con los siguientes servicios de AWS:

- Amazon S3.
- Amazon MSK.
- Amazon OpenSearch Service.
- Amazon DynamoDB.
- Amazon Kinesis Data Streams.
- Amazon Kinesis Data Firehose.
- Amazon CloudWatch.
- AWS Glue Schema Registry.

Capacidades avanzadas de integración

Además de las **integraciones** de AWS, las bibliotecas de Amazon Managed Service para Apache Flink incluyen más de cuarenta conectores de Apache Flink y la capacidad de crear integraciones personalizadas. Con algunas líneas de código más, puede modificar el comportamiento de cada integración con la funcionalidad avanzada. También puede crear integraciones personalizadas mediante un conjunto de tipos primitivos de Apache Flink que le permiten leer y escribir desde archivos, directorios, conectores u otros orígenes a las que puede acceder a través de Internet.

- **Procesamiento único.** Con Amazon Managed Service para Apache Flink, puedes crear aplicaciones en las que los registros procesados afectan los resultados exactamente una vez, lo que se conoce como procesamiento único. Incluso en el caso de una interrupción de la aplicación, como el mantenimiento del servicio interno o la actualización de la aplicación iniciada por el usuario, el servicio garantiza que todos los datos se procesen y que no haya datos duplicados.

- **Procesamiento con estado.** El servicio almacena el procesamiento o estado previo y en curso en el almacenamiento de la aplicación en ejecución. Compara los resultados pasados y actuales durante cualquier período de tiempo y logra una recuperación rápida durante las interrupciones de la aplicación. El estado siempre está cifrado y se guarda de manera progresiva en el almacenamiento de la aplicación en ejecución.

- **Copias de seguridad de aplicaciones duraderas.** Se puede crear y eliminar copias de seguridad de aplicaciones duraderas a través de una simple llamada a la API. Restaura inmediatamente sus aplicaciones desde la última copia de seguridad después de una interrupción o su aplicación a una versión anterior.

- **Integración de ML.** Amazon Managed Service para Apache Flink admite algoritmos de aprendizaje automático (ML). Puedes crear aplicaciones en tiempo real para la clasificación, la agrupación en clústeres, la evaluación, las recomendaciones de ingeniería de características, las regresiones y las estadísticas.

- **Compatibilidad con el registro de esquemas de AWS Glue.** Amazon Managed Service para Apache Flink es compatible con AWS Glue Schema Registry. Schema Registry permite mejorar la calidad de los datos y protegerse frente a cambios inesperados mediante comprobaciones de compatibilidad que rigen la evolución de los esquemas en Amazon Managed Service para Apache Flink para cargas de trabajo conectadas a Apache Kafka, Amazon MSK o Amazon Kinesis Data Streams, ya sea como conector de origen o de destino.

Tema 2. Ingesta de datos para big data e inteligencia artificial en cloud

2.4. Referencias bibliográficas

Aitor Medrano. (s. f.). *Ingesta de datos*. GitHub. https://aitor-medrano.github.io/iabd/de/etl.html

Amazon Web Services. (2016, septiembre 8). *Introduction to Amazon Kinesis Firehose* [Vídeo]. YouTube. https://youtu.be/8L3lLSPPxpY

Amazon Web Services. (2020, julio 2). *Amazon Kinesis Data Streams Fundamentals* [Vídeo]. YouTube. https://youtu.be/hLLgkTUmwOU

Amazon Web Services. (2022, agosto 23). *AWS Glue Overview | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/u14iVEc-C6E

Amazon Web Services. (2023a, julio 18). *AWS Glue Studio - Visual data pipeline demo | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/ckxwnd4BQmk

Amazon Web Services. (2023b, febrero 17). *Introduction to Kinesis Data Firehose | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/qRoyF9dEqgw

Amazon Web Services. (2023c, agosto 30). *Introduction to Amazon Managed Service for Apache Flink | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/vI1GiMSHuxM

Amazon Web Services. (2023d, noviembre 29). *AWS re:Invent 2017 - Introducing Amazon Kinesis Vídeo Streams* [Vídeo]. YouTube. https://youtu.be/STEMa3t5NOQ

Amazon Web Services. (2023e). *Data Consumer Options for Amazon Kinesis Data Streams | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/opY-56_9AOg

Amazon Web Services. (2023f, febrero 14). *Data Producer Options for Amazon Kinesis Data Streams | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/tmhCRGV0XOM

Amazon Web Services. (2023g, enero 25). *Getting Started with Kinesis Data Streams | Amazon Web Services* [Vídeo]. YouTube. https://youtu.be/1I1DcJvmd4w

Amazon Web Services. (2023h, octubre 16). One-Click Streaming with Amazon Managed Service For Apache Flink Blueprints | Amazon Web Services [Vídeo]. YouTube. https://youtu.be/mWTnArI8xCI

Amazon Web Services. (2024). *Developer Guide. Amazon Kinesis Data Streams*. https://docs.aws.amazon.com/pdfs/streams/latest/dev/kinesis-dg.pdf

Amazon Web Services. (s. f.-a). *Catalog and search*. https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/data-cataloging.html

Amazon Web Service. (s. f.-b). *Información general de los flujos de trabajo en AWS Glue*. https://docs.aws.amazon.com/es_es/glue/latest/dg/workflows_overview.html

Amazon Web Service. (s. f.-c). *Amazon Kinesis*. Fuente: https://aws.amazon.com/es/kinesis/

Amazon Web Service. (s. f.-d). *Amazon Data Firehose*. Fuente: https://aws.amazon.com/es/firehose/

Amazon Web Services. (s. f.-e). *AWS Documentation*. https://docs.aws.amazon.com/es_es/

Amazon Web Services. (s. f.-f). *Componentes de AWS Glue.* https://docs.aws.amazon.com/es_es/glue/latest/dg/components-overview.html#data-catalog-intro

Amazon Web Services. (s. f.-g). *Creación de trabajos de ETL visuales con AWS Glue Studio*. https://docs.aws.amazon.com/es_es/glue/latest/dg/author-job-glue.html

Tema 2. Ingesta de datos para big data e inteligencia artificial en cloud

Amazon Web Services. (s. f.-h). *Detección y catalogación de datos en AWS Glue*. https://docs.aws.amazon.com/es_es/glue/latest/dg/catalog-and-crawler.html

Amazon Web Services. (s. f.-i). *Documentación de Amazon Kinesis.* https://docs.aws.amazon.com/es_es/kinesis/

Amazon Web Services. (s. f.-j). *¿Qué es Amazon Kinesis Data Streams?* https://docs.aws.amazon.com/es_es/streams/latest/dev/introduction.html

Amazon Web Services. (s. f.-k). *¿Qué es Amazon Data Firehose?* https://docs.aws.amazon.com/es_es/firehose/latest/dev/what-is-this-service.html

Amazon Web Services. (s. f.-l). *¿Qué es Amazon Managed Service para Apache Flink?* https://docs.aws.amazon.com/es_es/managed-flink/latest/java/what-is.html

AWS LATAM. (2021a, diciembre 6). *Descubre y cataloga tus datos con AWS Glue – Español* [Vídeo]. YouTube. https://youtu.be/pbKnfjjVsx4

AWS LATAM. (2021b, diciembre 6). *Prepara tus datos y administra tus ETLs con AWS Glue - Español* [Vídeo]. YouTube. https://youtu.be/mw6nu7-_4PI

AWS Tutorials. (2022, marzo 7). *AWS Tutorials - Interactively Develop Glue Job using Jupyter Notebook* [Vídeo]. YouTube. https://youtu.be/n4PVC5O_tJo

Blokdyk, G. (2021). *AWS Glue Second Edition*. 5STARCooks.

Development Team. (2018). *Amazon Kinesis Firehose Developer Guide*. Samurai Media Limited.

Halder, N. (2023, mayo 27). *ETL vs. ELT: A Comprehensive Comparison and Guide to Modern Data Integration Strategies*. Medium. https://medium.com/analysts-corner/etl-vs-elt-a-comprehensive-comparison-and-guide-to-modern-data-integration-strategies-f2968bc64651

IMPACT Unofficial. (2022, noviembre 4 *). AI Foundation Models to Augment Scientific Data and the Research Lifecycle*. Medium. https://impactunofficial.medium.com/ai-foundation-models-to-augment-scientific-data-and-the-research-lifecycle-d5c16054df89

Kumar V, S. (2023). *Stream Processing with Apache Flink & Pyflink: A Comprehensive Guide* [autoedición].

Makota, T., Maguire, B., Gagne, D. y Chakrabarti, R. (2023). *Scalable Data Streaming with Amazon Kinesis: Design and secure highly available, cost-effective data streaming applications with Amazon Kinesis*. Packt Publishing.

Murat Sivri. (2022, septiembre 7). *Batch processing - stream processing*. Medium. https://medium.com/i%CC%87stanbuldatascienceacademy/batch-processing-stream-processing-74a88a7c0dc7

Saad, A., Salem, R. y Abd eldkader, H. A. (2021). Frequent Pattern Mining over Streaming Data: From models to research challenges. *IJCI International Journal of Computers and Information, 8*(2), 156-161. https://www.researchgate.net/publication/356996515_Frequent_Pattern_Mining_over_Streaming_Data_From_models_to_research_challenges

The Art of Service - AWS Glue Publishing (Author). (2021). *AWS Glue A Complete Guide*.

Qlik. (s. f.). *Streaming Data*. https://www.qlik.com/us/streaming-data

Vrinda Mathur. (2022, junio 21). *What is Data Ingestion? Challenges and Types*. AnalyticSteps. https://www.analyticssteps.com/blogs/what-data-ingestion-challenges-and-types

Tema 2. Ingesta de datos para big data e inteligencia artificial en cloud

Waehner, K. (2022, junio 27). *Data Warehouse vs. Data Lake vs. Data Streaming – Friends, Enemies, Frenemies?* Kai Waehner. https://www.kai-waehner.de/blog/2022/06/27/data-warehouse-vs-data-lake-vs-data-streaming-friends-enemies-frenemies/

Wall Street Mindset. (2024, enero 13). *Automated ETL Workflow Orchestration with AWS Glue, Athena, Lambda, EventBridge, and Step Functions* [Vídeo]. YouTube. https://youtu.be/Olt0XekklhU

Tema 3. Data science cloud storage

3.1. Introducción y objetivos

Siguiendo con las etapas del ciclo de vida del dato en el ámbito de *big data* y *data science,* y la etapa de ingesta del dato, entramos a revisar los principales conceptos, paradigmas y tecnologías en el ámbito del almacenamiento para la ciencia del dato.

Los **objetivos** para este tema son:

- Entender y diferenciar los conceptos de *data lake*, *data warehouse* y *data lakehouse*.
- Entender los nuevos paradigmas de arquitecturas de almacenamiento de datos.
- Comprender el paradigma *data mesh* y aprender sus elementos clave.
- Comprender el paradigma *data fabric* y aprender sus elementos clave.
- Conocer y aprender las principales características y funcionalidades del servicio de almacenamiento masivo AWS S3.
- Conocer y aprender las principales características y funcionalidades del servicio de almacenamiento HDFS.
- Conocer y aprender las principales características y funcionalidades del servicio de *data warehouse* en la nube de AWS Redshift.

3.2. Sistemas de almacenamiento: big data

Sistemas de almacenamiento

Imagina un sistema de almacenamiento de datos como un gigantesco almacén donde guardas todas tus pertenencias. Ahora, en lugar de ropa, muebles o recuerdos personales, este almacén guarda datos, por ejemplo, desde simples números hasta imágenes, vídeos, documentos y mucho más. Por tanto, ¿qué **características** debe tener un sistema de almacenamiento?

- **Propósito.** Un sistema de almacenamiento de datos existe para mantener registros de información valiosa para una organización o individuo. Esta información puede ser cualquier cosa, desde datos transaccionales en una empresa hasta fotos familiares en una nube personal.

- **Estructura.** Al igual que en un almacén físico, un sistema de almacenamiento de datos tiene una estructura organizada. Esto puede ser en forma de bases de datos, sistemas de archivos o incluso una combinación de ambos. Estas estructuras están diseñadas para almacenar y organizar datos de manera eficiente, permitiendo un fácil acceso y manipulación cuando sea necesario.

- **Tipos de datos.** Los sistemas de almacenamiento de datos pueden manejar una amplia variedad de tipos de datos, desde simples cadenas de texto hasta datos complejos, como imágenes, vídeos, registros de transacciones financieras, datos biométricos, etc. Dependiendo de la naturaleza de los datos que se están almacenando, el sistema puede estar optimizado para manejar diferentes tipos de carga de trabajo y consultas.

- **Escalabilidad.** Uno de los desafíos clave en el diseño de un sistema de almacenamiento de datos es la escalabilidad. A medida que la cantidad de datos aumenta con el tiempo, el sistema debe ser capaz de crecer y manejar esta carga adicional sin comprometer el rendimiento. Esto puede implicar la adición de más

servidores, el uso de tecnologías de almacenamiento distribuido o la migración a soluciones en la nube.

- **Seguridad.** Dado que los datos almacenados son a menudo valiosos y sensibles, la seguridad es una consideración crítica en cualquier sistema de almacenamiento de datos. Esto implica medidas como el cifrado de datos, el control de acceso basado en roles, la monitorización de la actividad del usuario y la implementación de políticas de retención de datos.

- **Recuperación y copia de seguridad.** Los sistemas de almacenamiento de datos también deben estar preparados para situaciones de pérdida de datos, ya sea por error humano, fallos de *hardware* o ciberataques. Por lo tanto, suelen incluir capacidades de copia de seguridad y recuperación para garantizar la integridad y disponibilidad de los datos en todo momento.

Por tanto, el **sistema de almacenamiento de datos** es un repositorio central donde se guarda información para que se pueda consultar, analizar o visualizar para tomar decisiones mejor informadas. Los datos fluyen hacia un almacenamiento de datos desde sistemas transaccionales, IoT, *logs*, bases de datos relacionales y otros orígenes. Los analistas empresariales, los ingenieros de datos, los científicos de datos y los responsables de la toma de decisiones obtienen acceso a los esos repositorios de almacenamiento a través de diferentes herramientas.

Los **datos** y el **análisis** se han vuelto fundamentales para que las empresas mantengan la competitividad. Las empresas utilizan informes, *dashboards* y herramientas de análisis para extraer información de los datos, monitorear el desempeño de la empresa y respaldar la toma de decisiones.

En resumen, un sistema de almacenamiento de datos es la **infraestructura subyacente** que permite a las organizaciones y personas almacenar, organizar, acceder y proteger grandes volúmenes de información de manera eficiente y segura, por lo que desempeña un papel fundamental en la gestión de datos en la era digital.

Concepto de almacenamiento en *big data*

El **almacenamiento** en *big data* es un componente fundamental en la gestión y el análisis de grandes volúmenes de datos. Este concepto se refiere a la capacidad de almacenar, gestionar y acceder a enormes cantidades de datos de manera eficiente y escalable. En el contexto de *big data*, el almacenamiento debe ser capaz de manejar datos estructurados, no estructurados y semiestructurados, provenientes de diversas fuentes, como redes sociales, sensores, registros de transacciones, archivos de texto, imágenes, vídeos, entre otros.

A continuación, se explican algunos aspectos clave del almacenamiento en *big data*:

- **Escalabilidad.** El almacenamiento en *big data* debe ser altamente escalable para poder manejar la creciente cantidad de datos de manera efectiva. Esto implica la capacidad de agregar más capacidad de almacenamiento y procesamiento según sea necesario, sin comprometer el rendimiento.

- **Tolerancia a fallos.** Dado que los sistemas de *big data* suelen trabajar con enormes cantidades de datos distribuidos en múltiples nodos, es crucial que el almacenamiento sea tolerante a fallos. Esto significa que el sistema debe poder mantener la disponibilidad y la integridad de los datos incluso en caso de que uno o varios nodos fallen.

- **Distribución.** Los datos en *big data* se distribuyen en múltiples nodos o servidores para permitir un procesamiento paralelo y una mayor velocidad de acceso. El almacenamiento debe ser capaz de distribuir y gestionar estos datos de manera eficiente, lo que optimiza la carga de trabajo en toda la infraestructura.

- **Velocidad.** El acceso rápido a los datos es fundamental en el análisis de *big data*. Por lo tanto, el almacenamiento debe ser capaz de proporcionar altas velocidades de lectura y escritura, incluso cuando se trabaja con conjuntos de datos masivos.

- **Flexibilidad.** Dado que los datos en *big data* pueden tener diferentes estructuras y formatos, el almacenamiento debe ser flexible y compatible con una variedad de tipos de datos. Esto puede implicar el uso de bases de datos NoSQL, sistemas de archivos distribuidos, como HDFS, almacenamiento en la nube, entre otros.

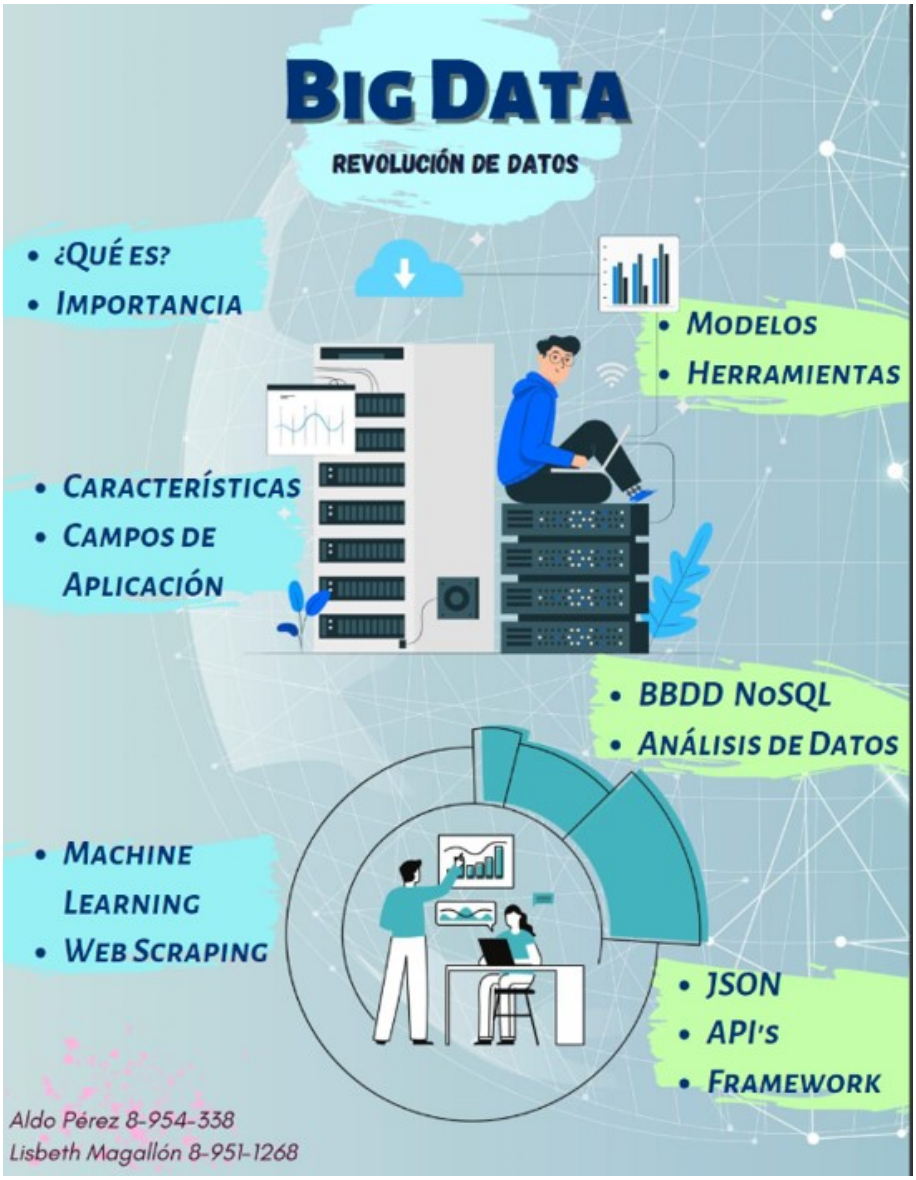

Figura 1. *Big data*. Fuente: LisM24, 2021.

En resumen, el almacenamiento en *big data* es esencial para la gestión eficiente de grandes volúmenes de datos, garantizando escalabilidad, tolerancia a fallos, distribución, velocidad y flexibilidad para satisfacer las demandas del análisis de datos a gran escala.

Paradigma de almacenamiento distribuido

El **paradigma de los sistemas de almacenamiento distribuido** en *big data* es una pieza fundamental en la gestión eficiente de grandes volúmenes de datos. Este enfoque surge como respuesta a los desafíos que enfrentan las organizaciones al intentar almacenar, procesar y analizar cantidades masivas de datos que superan la capacidad de los sistemas de almacenamiento tradicionales. Seguidamente se explica con detalle los conceptos, las características, el origen y la utilidad de este paradigma.

- **Concepto.** El sistema de almacenamiento distribuido en *big data* se refiere a una arquitectura en la que los datos se distribuyen y almacenan en múltiples nodos o servidores, en lugar de concentrarse en una única ubicación centralizada. Estos nodos pueden estar ubicados en diferentes lugares geográficos y se comunican entre sí a través de una red. Los datos se replican y fragmentan en estos nodos para permitir un procesamiento paralelo y una mayor tolerancia a fallos.

- **Características.** Los sistemas de almacenamiento distribuido pueden escalar horizontalmente agregando más nodos al sistema, lo que permite manejar grandes volúmenes de datos sin sacrificar el rendimiento.

  - **Tolerancia a fallos.** Al distribuir los datos en múltiples nodos, el sistema puede tolerar la falla de uno o varios nodos sin perder datos o interrumpir el servicio. La replicación de datos y la redundancia juegan un papel clave en este aspecto.

  - **Procesamiento paralelo.** La distribución de datos permite que múltiples nodos

procesen los datos de forma simultánea, lo que acelera el procesamiento y el análisis de grandes conjuntos de datos.

- **Alto rendimiento.** Gracias al procesamiento paralelo y la escalabilidad horizontal, los sistemas de almacenamiento distribuido pueden ofrecer un alto rendimiento en términos de velocidad de acceso y procesamiento de datos.

- **Flexibilidad.** Estos sistemas son flexibles y pueden manejar una amplia variedad de tipos de datos, desde datos estructurados hasta datos no estructurados y semiestructurados.

- **Origen.** El surgimiento del paradigma de almacenamiento distribuido en *big data* está estrechamente relacionado con el crecimiento exponencial de datos en la era digital. Con la proliferación de dispositivos conectados, redes sociales, sensores y otras fuentes de datos, las organizaciones se enfrentaron al desafío de gestionar y analizar enormes cantidades de información de manera eficiente. Los sistemas de almacenamiento distribuido surgieron como una solución escalable y flexible para abordar estos desafíos. Además, el concepto de almacenamiento distribuido en *big data* no tiene un creador único, sino que surge como resultado de la evolución de varias tecnologías y enfoques a lo largo del tiempo. Sin embargo, hay ciertos hitos y contribuciones importantes que han dado forma a este paradigma. Aquí hay algunos puntos clave:

- **Google File System (GFS).** Uno de los hitos fundamentales en el desarrollo de sistemas de almacenamiento distribuido fue el GFS, presentado por Google en un artículo de investigación en 2003. GFS fue diseñado para satisfacer las necesidades de almacenamiento de Google, lo que permitió el almacenamiento de grandes volúmenes de datos en clústeres de servidores.

- **MapReduce.** Otro avance crucial fue la introducción de MapReduce, también por Google en 2004. MapReduce es un modelo de programación y procesamiento distribuido diseñado para procesar grandes conjuntos de datos en paralelo en un clúster de servidores. Este modelo se convirtió en la base de muchas tecnologías de

big data posteriores.

- **Hadoop.** Basándose en los principios establecidos por GFS y MapReduce, Doug Cutting y Mike Cafarella crearon Hadoop, en 2005, como un proyecto de código abierto inspirado en los trabajos de Google. Hadoop consiste en un sistema de archivos distribuido llamado HDFS y un marco de procesamiento distribuido basado en MapReduce. Hadoop fue fundamental en popularizar el concepto de almacenamiento distribuido en el ámbito de *big data*.

- **Apache Spark.** Aunque Hadoop fue un avance significativo, tenía algunas limitaciones en términos de rendimiento y flexibilidad. Apache Spark surgió como una alternativa más rápida y versátil al modelo MapReduce. Desarrollado en la Universidad de California, Berkeley, y lanzado como proyecto de código abierto en 2010, Apache Spark permite un procesamiento de datos distribuido en memoria, lo que lo hace más adecuado para aplicaciones de análisis en tiempo real y aprendizaje automático.

- **Otras tecnologías.** Además de Hadoop y Spark, han surgido muchas otras tecnologías de almacenamiento y procesamiento distribuido en el ecosistema de *big data*. Estas incluyen sistemas de bases de datos NoSQL, como Cassandra y MongoDB, sistemas de procesamiento de transmisiones, como Apache Kafka, sistemas de almacenamiento en la nube, como Amazon S3 y Google Cloud Storage, entre otros.

En resumen, el concepto de **almacenamiento distribuido en *big data*** surge como resultado de la combinación de varios avances tecnológicos y contribuciones de investigación, con hitos clave como el Google File System, MapReduce, Hadoop y Apache Spark. Estas tecnologías han sentado las bases para la gestión eficiente de grandes volúmenes de datos en entornos distribuido.

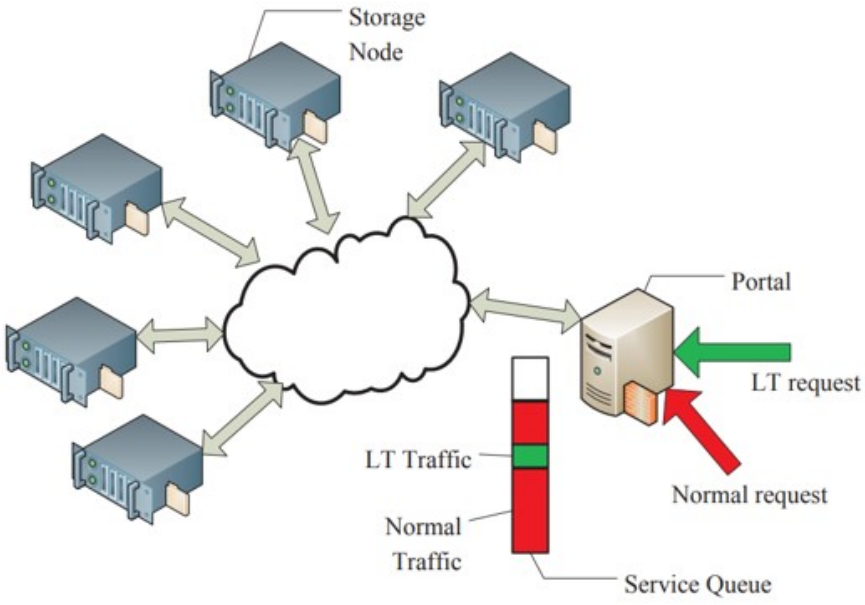

Figura 2. Arquitectura de un sistema de almacenamiento distribuido de datos. Fuente: Lu et al., 2012.

- **Utilidad.** El paradigma de almacenamiento distribuido en *big data* es esencial para una variedad de aplicaciones y casos de uso, incluyendo análisis de datos en tiempo real:
  - Procesamiento de datos masivos para informes y análisis.
  - Almacenamiento y análisis de registros de aplicaciones y sistemas.
  - Aplicaciones de inteligencia artificial y aprendizaje automático que requieren acceso a grandes conjuntos de datos.
  - Análisis de datos en la nube y entornos de computación distribuida.

En resumen, el **paradigma** de almacenamiento distribuido en *big data* proporciona una infraestructura robusta y escalable para almacenar y procesar grandes volúmenes de datos, lo que les permite a las organizaciones extraer información valiosa y tomar decisiones basadas en datos en un entorno cada vez más digitalizado y complejo.

Tipos de almacenamiento de datos

Como hemos visto, el almacenamiento de datos es fundamental en la gestión de la información en entornos empresariales y de computación en la nube. Cada tipo de almacenamiento tiene sus propias características y se adapta mejor a diferentes tipos de datos y aplicaciones. Comprender las diferencias entre ellos es crucial para tomar decisiones informadas sobre cómo almacenar y gestionar datos de manera eficiente y efectiva. Existen tres **tipos principales de almacenamiento de datos:** almacenamiento de objetos, almacenamiento de archivos y almacenamiento de bloques.

E l **almacenamiento de objetos** es un tipo de almacenamiento de datos que organiza la información en forma de objetos individuales, donde cada objeto contiene datos, metadatos y un identificador único. Los objetos se almacenan en un repositorio centralizado y se accede a ellos a través de una interfaz de programación de aplicaciones (API) estándar. Las **características** clave del almacenamiento de objetos incluyen:

- **Escalabilidad.** Puede manejar grandes volúmenes de datos y escalar horizontalmente según sea necesario.
- **Flexibilidad.** Admite una amplia variedad de tipos de datos, desde documentos e imágenes hasta vídeos y archivos de audio.
- **Durabilidad.** Proporciona una alta durabilidad de datos mediante la replicación y la distribución de objetos en múltiples ubicaciones.
- **Acceso mediante API.** Los objetos se acceden a través de una API estándar, lo que facilita su integración con aplicaciones y sistemas existentes.

Por otro lado, el **almacenamiento de archivos** organiza la información en forma de archivos individuales, donde cada archivo tiene una ubicación y una estructura definida en el sistema de archivos. Los archivos se almacenan en un sistema de archivos jerárquico y se accede a ellos a través de rutas de acceso. Las **características** clave del almacenamiento de archivos incluyen:

- **Organización jerárquica.** Los archivos se organizan en una estructura de directorios y subdirectorios, lo que facilita la organización y la gestión de datos.
- **Acceso basado en rutas de acceso.** Los archivos se acceden a través de rutas de acceso, como URL o rutas de archivos locales, que especifican su ubicación en el sistema de archivos.
- **Compatibilidad con sistemas de archivos estándar.** Admite sistemas de archivos estándar, como NTFS en Windows y ext4 en Linux, lo que facilita la integración con los sistemas operativos y las aplicaciones existentes.
- **Eficiencia en la transferencia de archivos.** Permite la transferencia eficiente de archivos grandes mediante técnicas como la compresión y el almacenamiento en caché.

Por último, el **almacenamiento de bloques** organiza la información en forma de bloques de datos individuales, donde cada bloque tiene un identificador único y se almacena en un dispositivo de almacenamiento físico, como un disco duro o una unidad de estado sólido (SSD). Los bloques se acceden a través de un sistema de gestión de almacenamiento y se utilizan para construir archivos y sistemas de archivos. Las **características** clave del almacenamiento de bloques incluyen:

- **Eficiencia en el almacenamiento de datos.** Permite un uso eficiente del espacio de almacenamiento mediante la asignación de bloques de datos en función de las necesidades de cada archivo.

- **Rendimiento optimizado.** Proporciona un rendimiento óptimo para operaciones de lectura y escritura, especialmente en entornos de almacenamiento de alta velocidad, como servidores de bases de datos y sistemas de almacenamiento de archivos.
- **Seguridad y protección de datos.** Proporciona mecanismos de seguridad y protección de datos, como la encriptación y la replicación de datos, para garantizar la integridad y la disponibilidad de la información almacenada.

En resumen, el almacenamiento de objetos, archivos y bloques son tres tipos principales de **almacenamiento de datos,** cada uno con sus propias características y casos de uso. Comprender las diferencias entre ellos es crucial para tomar decisiones informadas sobre cómo almacenar y gestionar datos de manera eficiente y efectiva en entornos empresariales y de computación en la nube.

Figura 3. Tipos de almacenamiento. Fuente: Stackscale, 2023.

Almacenamiento por niveles

El paradigma de **almacenamiento por niveles,** también conocido como jerarquía de almacenamiento, es una estrategia utilizada en la gestión de datos para optimizar el rendimiento y la eficiencia en el acceso y almacenamiento de información. Este enfoque implica organizar los datos en diferentes niveles o capas de almacenamiento, donde cada nivel tiene características y costos diferentes. A continuación, tienes una explicación detallada.

El almacenamiento por niveles se basa en la premisa de que no todos los datos son igualmente importantes ni requerirán el mismo nivel de acceso o rendimiento en todo momento. Por lo tanto, los datos se organizan en varias **capas de almacenamiento,** desde niveles de almacenamiento rápidos y costosos hasta niveles más lentos, pero más económicos. Muy relacionado con el sistema de almacenamiento por niveles, es el concepto de temperatura del dato, que se explica a continuación.

L a **temperatura del dato** es un concepto que se utiliza en el ámbito del almacenamiento de datos para referirse a la frecuencia y la recencia con la que se accede a un determinado conjunto de datos. De manera similar al término «calor» en física, que indica la actividad o la importancia de una entidad, la temperatura del dato se refiere a la actividad o el nivel de uso de un conjunto de datos en particular. La relación entre la temperatura del dato y el sistema de almacenamiento por niveles es fundamental para optimizar el rendimiento y los costos en la gestión de datos. Las principales temperaturas del dato que se distinguen son la siguiente:

- **Datos calientes (*hot data*).** Se refiere a los datos que se acceden con frecuencia o recientemente. Estos datos suelen ser críticos para las operaciones en curso y necesitan un acceso rápido y eficiente.
- **Datos tibios (*warm data*).** Son datos que se acceden ocasionalmente pero que aún tienen cierta relevancia y pueden necesitar un acceso rápido en determinadas circunstancias.

- **Datos fríos (*cold data*).** Son datos que rara vez se acceden, generalmente son datos históricos o de archivo que se mantienen por motivos de cumplimiento normativo o para referencia futura.

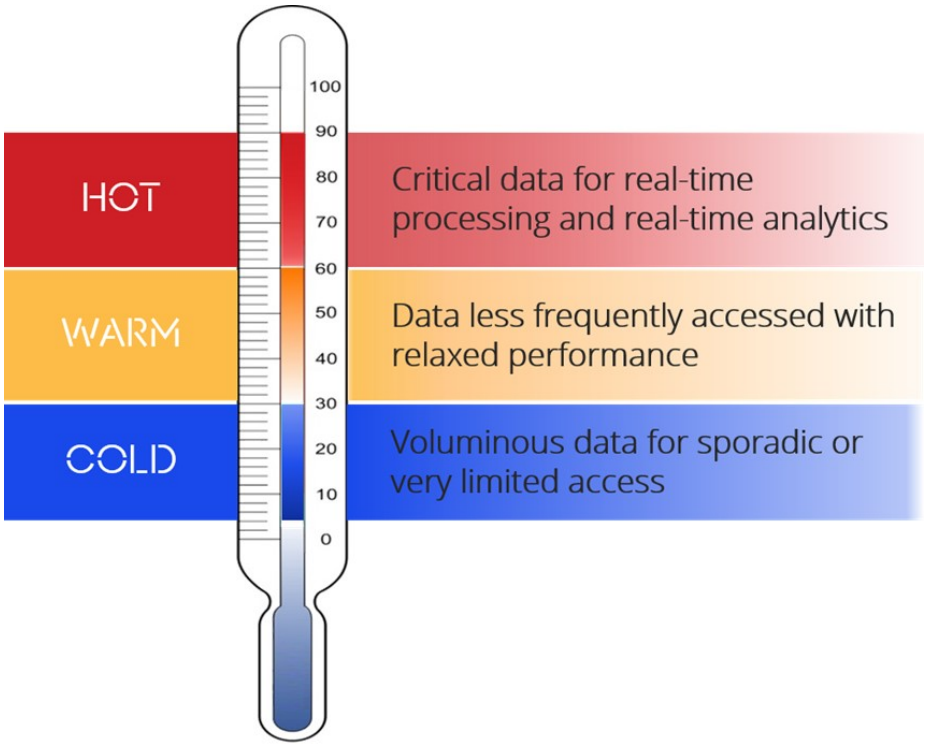

Figura 4. Temperaturas del dato. Fuente: SAPTOOLS, s. f.

En cuanto a la **relación con el sistema de almacenamiento por niveles,** el concepto de temperatura del dato se integra perfectamente con la estrategia de almacenamiento por niveles para optimizar el rendimiento y los costos en la gestión de datos. Seguidamente, profundizamos cómo se relacionan:

- **Asignación de datos a niveles de almacenamiento adecuados:**

  - Los **datos calientes** se colocan en niveles de almacenamiento de alta velocidad y baja latencia, como la memoria RAM y los discos de estado sólido (SSD), para garantizar un acceso rápido y eficiente.

- Los **datos tibios** pueden colocarse en niveles de almacenamiento de capacidad media, como discos duros tradicionales (HDD).
- Los **datos fríos** se pueden almacenar en niveles de almacenamiento de gran capacidad, pero de menor velocidad, como almacenamiento en cinta o almacenamiento en la nube de bajo costo.

- **Movimiento dinámico de datos entre niveles.** Los sistemas de almacenamiento por niveles pueden monitorear continuamente la actividad de acceso a los datos y mover dinámicamente los datos entre los diferentes niveles de almacenamiento según su temperatura. Por ejemplo, si un conjunto de datos tibios comienza a experimentar un aumento en su actividad de acceso, el sistema puede moverlo automáticamente a un nivel de almacenamiento más rápido para garantizar un rendimiento óptimo.

- **Reducción de costes.** Al asignar los datos a niveles de almacenamiento adecuados según su temperatura, el almacenamiento por niveles puede ayudar a reducir los costos generales de almacenamiento al utilizar dispositivos de almacenamiento más económicos para almacenar datos menos activos.

En resumen, la **temperatura del dato** y el **sistema de almacenamiento por niveles** están estrechamente relacionados en la gestión eficiente de datos, lo que permite una asignación óptima de recursos de almacenamiento según la actividad y la importancia de los datos en un entorno de almacenamiento de datos cada vez más complejo y diverso.

Respecto a un ejemplo de **infraestructura** típica de sistema de almacenamiento por niveles, podemos describir la siguiente:

- **Nivel 1 (almacenamiento de alta velocidad).** Este nivel incluye dispositivos de almacenamiento de alta velocidad y baja latencia, como la memoria RAM y los SSD. Se utilizan para almacenar datos que necesitan un acceso rápido y frecuente, como los datos en caché y los conjuntos de datos activos.

- **Nivel 2 (almacenamiento de capacidad media).** Este nivel consiste en dispositivos de almacenamiento con una capacidad mayor, pero un rendimiento menor en comparación con el nivel 1. Esto puede incluir discos duros tradicionales (HDD) u otros medios de almacenamiento de capacidad moderada. Se utilizan para almacenar datos que se acceden con menos frecuencia pero que aún necesitan estar disponibles de manera rápida cuando sea necesario.

- **Nivel 3 (almacenamiento de gran capacidad).** Este nivel implica dispositivos de almacenamiento de alta capacidad, pero con una velocidad de acceso más baja en comparación con los niveles anteriores. Esto puede incluir almacenamiento en cinta, almacenamiento en la nube de acceso esporádico o incluso almacenamiento en disco de baja velocidad. Se utilizan para almacenar datos de archivo o de respaldo que se acceden con poca frecuencia pero que aún necesitan estar disponibles para recuperación o referencia.

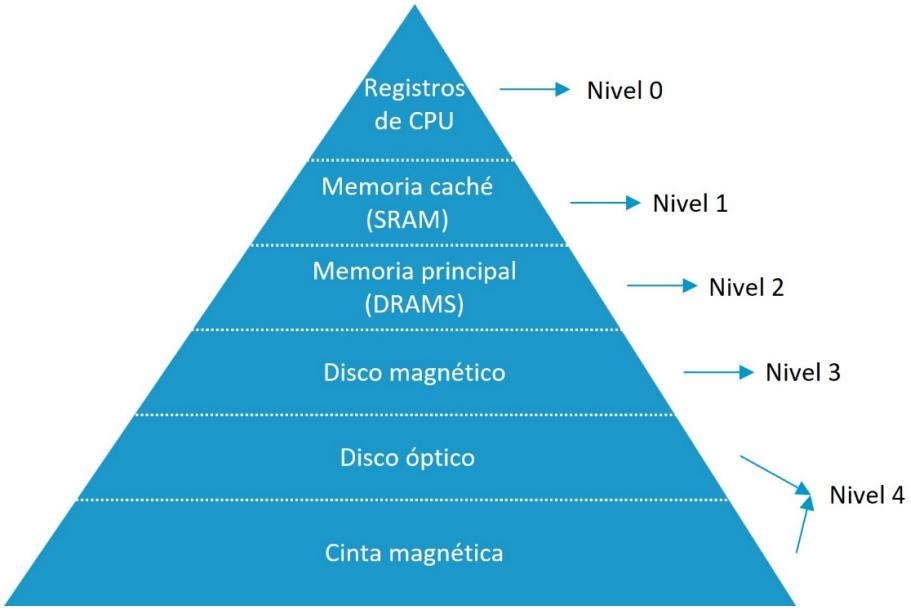

Figura 5. Jerarquía de almacenamiento. Fuente: elaboración propia.

El almacenamiento por niveles se utiliza en una variedad de entornos y aplicaciones, incluyendo:

- **Sistemas de gestión de bases de datos.** Los sistemas de gestión de bases de datos pueden utilizar el almacenamiento por niveles para optimizar el rendimiento de las consultas y las transacciones, lo que coloca los datos más activos en los niveles de almacenamiento más rápidos y costosos.
- **Almacenamiento de datos empresariales.** Las empresas pueden implementar almacenamiento por niveles para administrar eficientemente grandes volúmenes de datos, lo que asegura que los datos críticos estén disponibles de manera rápida y que los datos menos activos se almacenen de manera rentable.
- **Aplicaciones de almacenamiento en la nube.** Los proveedores de servicios en la nube pueden utilizar el almacenamiento por niveles para optimizar la asignación de recursos y los costos, lo que coloca a los datos de los clientes en diferentes niveles de almacenamiento según sus necesidades y patrones de acceso.

Los **beneficios** son:

- **Optimización del rendimiento.** Al colocar los datos más activos en niveles de almacenamiento más rápidos, el almacenamiento por niveles puede mejorar significativamente el rendimiento de las aplicaciones y consultas.
- **Reducción de costos.** Al utilizar dispositivos de almacenamiento más económicos para almacenar datos menos activos, el almacenamiento por niveles puede ayudar a reducir los costos generales de almacenamiento.
- **Mejora de la eficiencia.** Al distribuir los datos en diferentes niveles de almacenamiento según su importancia y frecuencia de acceso, el almacenamiento por niveles puede mejorar la eficiencia en la gestión de datos y recursos de almacenamiento.

En resumen, el paradigma de almacenamiento por niveles es una estrategia efectiva para optimizar el rendimiento, reducir costos y mejorar la eficiencia en la gestión de datos en una variedad de entornos y aplicaciones.

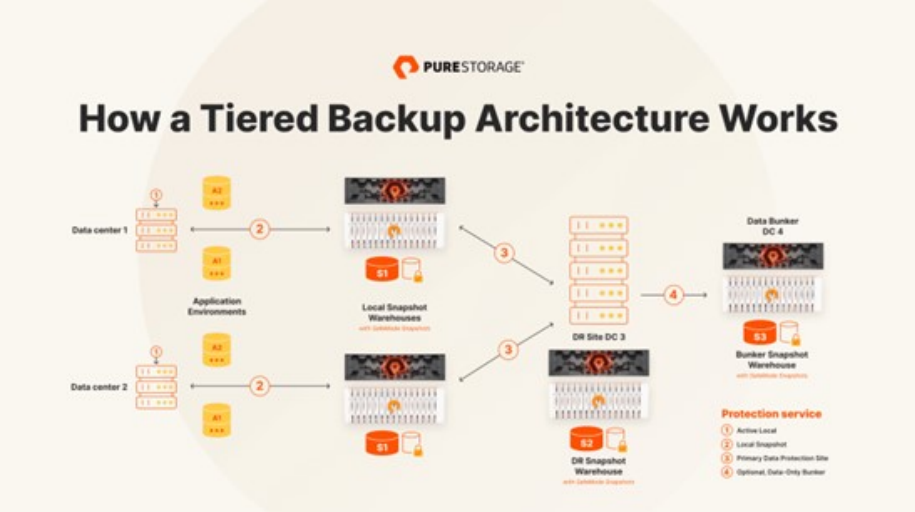

Figura 6. Clases de datos. Fuente: Pure Storage, s. f.

Arquitecturas de repositorios de datos: data lake, data warehouse, y data lakehouse

Los repositorios de datos son sistemas diseñados para almacenar, gestionar y proporcionar acceso a conjuntos de datos. Hay varios enfoques de repositorios de datos que varían en su estructura, características y casos de uso. En los siguientes apartados, vamos a profundizar en los principales enfoques existentes.

Data lake

U n ***data lake*** es un repositorio de datos centralizado que almacena grandes volúmenes de datos en su forma original, sin procesar, en una variedad de formatos y estructuras. A diferencia de los sistemas de almacenamiento tradicionales, como las bases de datos relacionales, que requieren que los datos estén estructurados

antes de almacenarlos, un *data lake* permite almacenar datos de cualquier tipo, incluidos datos estructurados, semiestructurados y no estructurados. Estos datos pueden incluir archivos de texto, imágenes, vídeos, registros de eventos, datos de sensores, datos de redes sociales y mucho más.

Entre sus principales **características,** se pueden citar:

- **El uso de múltiples herramientas y productos.** Un *data lake* cuenta con una serie de herramientas y productos que potencian su almacenaje y gestión de datos. En efecto, los más importantes forman parte de las siguientes características a mencionar, como, por ejemplo, el acceso y la interacción de los usuarios con los datos en remoto.

- **Capacidad de almacenamiento original.** Los datos se almacenan en su forma original, sin procesar ni estructurar previamente, lo que proporciona una mayor flexibilidad y diversidad de datos.

- **Almacenamiento seguro y catalogación de los datos.** Los *data lakes* le permiten almacenar datos relacionales, como los que surgen de bases de datos operativas y datos de aplicaciones de línea de negocio, y datos no relacionales, como los provenientes de aplicaciones móviles, dispositivos de IoT y redes sociales. También le brindan la capacidad de comprender qué datos hay en el *lake* a través del rastreo, la catalogación y la indexación de datos. Finalmente, los datos deben estar seguros para garantizar que sus activos de datos estén protegidos.

- **La gestión automatizada de metadatos.** Este factor se establece como una de las características de *data lake*s más destacables, puesto que este sistema de gestión automatizada permite que las actualizaciones de los metadatos se realicen de manera continua y programada. Además, esto te ahorrará tiempo y trabajo durante la gestión de metadatos.

- **La interacción de los usuarios con los datos.** Gracias a que *data lake* es un repositorio con acceso flexible y remoto, los usuarios interesados en la información

podrán acceder a esta desde diferentes partes y comprender la gestión de datos que van realizando. De esta manera, se otorga a un grupo de trabajo o empresa la posibilidad de implementarlo como una estrategia empresarial que logre una mejora en la toma de decisiones y las rutas de acción.

- **Los flujos de trabajo de ingestión configurables.** Dentro de las características de *data lake*s, esta es una de las que ofrece mayor flexibilidad, puesto que brinda una gran variedad de posibilidades para modificar cómo se desarrolla la ingesta. De esta manera, podrás establecer ciertos parámetros o dinámicas al flujo de trabajo de ingestión.

- **El *data lake* como un repositorio vivo.** Por otra parte, los *data lake*s se posicionan dentro del mundo *big data* gracias a su carácter de repositorio vivo, es decir, la posibilidad de gestionar y transformar los datos mientras estos se encuentran almacenados, a diferencia de otros sistemas, como *data warehouse*, que tarda mucho tiempo en llevar a cabo las modificaciones.

Entre sus principales **usos,** podemos destacar:

- **Análisis de *big data.*** Los *data lakes* son fundamentales para el análisis de grandes volúmenes de datos en entornos de *big data*, lo que permite que las organizaciones extraigan información valiosa y descubren patrones ocultos en los datos.

- **Exploración de datos.** Facilitan la exploración y el descubrimiento de datos, lo que les permite a los usuarios buscar, acceder y analizar datos de manera flexible y sin restricciones.

- **Integración de datos.** Permiten la integración de datos de diversas fuentes, incluyendo sistemas empresariales, aplicaciones web, dispositivos IoT, redes sociales y más.

- **Desarrollo de modelos de ML.** Son utilizados para entrenar y desarrollar modelos de ML y análisis predictivo, lo que proporciona un gran conjunto de datos para el aprendizaje automatizado.