Revista de sistemas de información inteligentes (2021) 56: 97–120
https://doi.org/10.1007/s10844-020-00608-7

Sobre arquitecturas de lagos de datos y gestión de metadatos

Pegdwende Sawadogo´ [1] **·Jer´ome Darmont** [1]

Recibido: 8 diciembre 2019 / Revisado: 20 marzo 2020 / Aceptado: 22 mayo 2020 /
Publicado en línea: 26 de enero de 2020

Resumen

En las últimas dos décadas, hemos sido testigos de un aumento exponencial de la producción de datos.
en el mundo. Los llamados grandes datos generalmente provienen de sistemas transaccionales, y aún más
así desde el Internet de las Cosas y las redes sociales. Se caracterizan principalmente por el volumen,
problemas de velocidad, variedad y veracidad. Los problemas relacionados con los grandes datos desafían fuertemente a los tradicionales
sistemas de gestión y análisis de datos. El concepto de lago de datos se introdujo para abordar
a ellos. Un lago de datos es un gran depósito de datos sin procesar que almacena y administra todos los datos de la empresa.
teniendo cualquier formato. Sin embargo, el concepto de lago de datos sigue siendo ambiguo o confuso para muchos
investigadores y profesionales, que a menudo la confunden con la tecnología Hadoop. Por lo tanto, nosotros
proporcionar en este documento un estado del arte integral de los diferentes enfoques para el lago de datos
diseño. Nos enfocamos particularmente en arquitecturas de lagos de datos y gestión de metadatos, que
son cuestiones clave en los lagos de datos exitosos. También discutimos los pros y los contras de los lagos de datos y
sus alternativas de diseño.

Palabras clave Lagos de datos · Arquitecturas de lagos de datos · Gestión de metadatos ·
Modelado de metadatos

1. Introducción

El siglo XXI está marcado por un crecimiento exponencial de la cantidad de datos producidos en
el mundo. Esto es inducido notablemente por el rápido desarrollo de la Internet de las Cosas (IoT)
y redes sociales Sin embargo, aunque los grandes datos representan una gran oportunidad para varias organizaciones,
vienen en tal volumen, velocidad, fuentes heterogéneas y estructuras que
superan las capacidades de los sistemas de gestión tradicionales para su recogida, almacenamiento y
procesamiento en un tiempo razonable (Miloslavskaya y Tolstoy 2016). Una solución probada en el tiempo
para la gestión y el procesamiento de big data es el almacenamiento de datos. Un almacén de datos es de hecho
un sistema de almacenamiento histórico e integrado que está específicamente diseñado para analizar datos.

Pegdwende Sawadogo
pegdwende.sawadogo@univ-lyon2.fr
Jer´ ome Darmont
jerome.darmont@univ-lyon2.fr

1 Universidad de Lyon, Lyon 2, ERIC UR 3083, Lyon, Francia

Sin embargo, mientras que los almacenes de datos siguen siendo relevantes y muy poderosos para los datos estructurados, los datos semiestructurados y no estructurados presentan grandes desafíos para los almacenes de datos. Sin embargo, la mayoría de los grandes datos no están estructurados (Miloslavskaya y Tolstoy 2016). Por lo tanto, se introdujo el concepto de lago de datos para abordar los problemas de big data, especialmente aquellos inducidos por la variedad de datos.

Un lago de datos es un sistema de análisis, gestión y almacenamiento de datos muy grande que maneja cualquier formato de datos. Actualmente es bastante popular y está de moda tanto en la industria como en la academia. Sin embargo, el concepto de lago de datos no es sencillo para todos. De hecho, una encuesta realizada en 2016 reveló que el 35 % de los encuestados consideraban los lagos de datos como una simple etiqueta de marketing para una tecnología preexistente, es decir, Apache Hadoop (Grosser et al. 2016).

Desde entonces, el conocimiento sobre el concepto del lago de datos ha evolucionado, pero todavía existen algunos conceptos erróneos, presumiblemente porque la mayoría de los enfoques de diseño de los lagos de datos son bocetos abstractos de la industria que brindan pocos detalles teóricos o de implementación (Quix y Hai 2018). Por lo tanto, una encuesta puede ser útil para brindar a los investigadores y profesionales una mejor comprensión del concepto de lago de datos y sus alternativas de diseño.

Hasta donde sabemos, las únicas revisiones bibliográficas sobre lagos de datos son bastante breves y/o se centran en un tema específico, por ejemplo, conceptos y definiciones de lagos de datos (Couto et al. 2019; Madera y Laurent 2016), las tecnologías utilizadas para implementar lagos de datos (Mathis 2017) o problemas inherentes a los lagos de datos (Giebler et al. 2019; Quix y Hai 2018).
Es cierto que el informe propuesto en Russom (2017) es bastante extenso, pero adopta una visión puramente industrial. Por lo tanto, adoptamos en este documento un alcance más amplio para proponer un estado del arte más completo de los diferentes enfoques para diseñar y explotar un lago de datos. Nos enfocamos particularmente en las arquitecturas de lagos de datos y la gestión de metadatos, que se encuentran en la base de cualquier proyecto de lagos de datos y son los temas más citados en la literatura (Fig. 1).

Más precisamente, primero revisamos las definiciones de los lagos de datos y complementamos la mejor existente. Luego, investigamos las arquitecturas y tecnologías utilizadas para la implementación de lagos de datos y proponemos una nueva tipología de arquitecturas de lagos de datos. Nuestro segundo enfoque principal es la gestión de metadatos, que es un tema primordial para evitar convertir un lago de datos en un pantano de datos inoperable. En particular, clasificamos los metadatos del lago de datos e introducimos las características necesarias para lograr un sistema completo de metadatos. También discutimos los pros y los contras de los lagos de datos.

Eventualmente, tenga en cuenta que no revisamos otros temas importantes, como la ingestión de datos, el gobierno de datos y la seguridad en los lagos de datos, porque actualmente se abordan poco en la literatura, pero presumiblemente aún podrían ser el tema de otra encuesta completa.

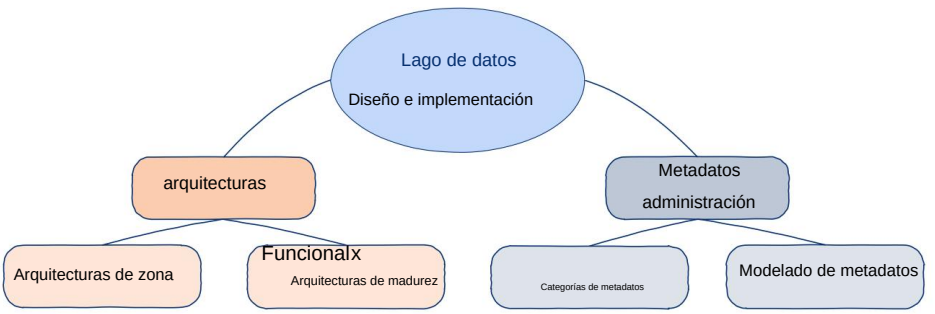


Fig. 1 Cuestiones abordadas en este documento

El resto de este documento está organizado de la siguiente manera. En la Sección 2, definimos el concepto de lago de datos. En la Sección 3, revisamos las arquitecturas y tecnologías de los lagos de datos para ayudar a los usuarios a elegir el enfoque y las herramientas correctos. En la Sección 4, revisamos y discutimos extensamente la gestión de metadatos. Finalmente, recapitulamos los pros y los contras de los lagos de datos en la Sección 5 y concluimos el artículo en la Sección 6 con un mapa mental de los conceptos clave que presentamos, así como los temas de investigación abiertos actuales.

2 Definiciones de lagos de datos

2.1 Definiciones de la literatura

Dixon introdujo el concepto de lago de datos como una solución a las deficiencias percibidas de los datamarts, que son subdivisiones específicas del negocio de los almacenes de datos que permiten responder solo subconjuntos de preguntas (Dixon 2010). En la literatura, los lagos de datos también se denominan reservorios de datos (Chessell et al. 2014) y centros de datos (Ganore 2015; Laskowski 2016), aunque los términos lago de datos son los más frecuentes. Dixon visualiza un lago de datos como un gran sistema de almacenamiento de datos heterogéneos sin procesar, alimentados por múltiples fuentes de datos, y que permite a los usuarios explorar, extraer y analizar los datos.

Posteriormente, parte de la literatura consideró a los lagos de datos como un equivalente a la tecnología Hadoop (Fang 2015; Ganore 2015; O'Leary 2014). De acuerdo con este punto de vista, el concepto de lago de datos se refiere a una metodología para utilizar tecnologías gratuitas o de bajo costo, típicamente Hadoop, para almacenar, procesar y explorar datos sin procesar dentro de una empresa (Fang 2015).

La asociación sistemática de lagos de datos a tecnologías de bajo costo se está volviendo minoritaria en la literatura, ya que el concepto de lago de datos ahora también se asocia con soluciones en la nube propietarias como Azure o IBM (Madera y Laurent 2016; Sirosh 2016) y varios sistemas de gestión de datos como como soluciones NoSQL y multitiendas. Sin embargo, todavía puede verse como un patrón de diseño basado en datos para la gestión de datos (Russom 2017).

De manera más consensuada, un lago de datos puede verse como un repositorio central donde los datos de todos los formatos se almacenan sin un esquema estricto, para análisis futuros (Couto et al. 2019; Khine y Wang 2017; Mathis 2017). Esta definición se basa en dos características clave de los lagos de datos: la variedad de datos y el enfoque de esquema en lectura, también conocido como enlace tardío (Fang 2015), lo que implica que los requisitos de esquema y datos no se fijan hasta que se consultan los datos (Khine y Wang 2017; Maccioni y Torlone 2018; Stein y Morrison 2014). Esto es lo opuesto al enfoque de esquema en escritura utilizado en los almacenes de datos.

Sin embargo, la definición de variedad/esquema en lectura puede considerarse confusa porque brinda pocos detalles sobre las características de un lago de datos. Por lo tanto, Madera y Laurent introducen una definición más completa en la que un lago de datos es una vista lógica de todas las fuentes y conjuntos de datos en su formato original, accesible a los científicos de datos o estadísticos para la extracción de conocimiento (Madera y Laurent 2016).

Más interesante aún, esta definición se complementa con un conjunto de características que un lago de datos debe incluir:

1. la calidad de los datos la proporciona un conjunto de metadatos; 2. el lago está controlado por herramientas de política de gobierno de datos; 3. el uso del lago está limitado a estadísticos y científicos de datos; 4. el lago integra datos de todo tipo y formato; 5. el lago de datos tiene una organización lógica y física.

2.2 Discusión y nueva definición

La definición de lagos de datos de Madera y Laurent es presumiblemente la más precisa, ya que define los requisitos que debe cumplir un lago de datos (Sección 2.1). Sin embargo, algunos puntos de esta definición son discutibles.

De hecho, los autores restringen el uso del lago a los especialistas en datos y, en consecuencia, excluyen a los expertos en negocios por razones de seguridad. Sin embargo, en nuestra opinión, es completamente posible permitir el acceso controlado a este tipo de usuarios a través de una capa de software de análisis o navegación.

Además, no compartimos la visión del lago de datos como una vista lógica sobre las fuentes de datos, ya que algunas fuentes de datos pueden ser externas a una organización y, por lo tanto, al lago de datos.
Dado que Dixon establece explícitamente que los datos del lago provienen de fuentes de datos (Dixon 2010), incluir fuentes de datos en el lago puede considerarse contrario al espíritu de los lagos de datos.

Finalmente, aunque bastante completa, la definición de Madera y Laurent omite una propiedad esencial de los lagos de datos: la escalabilidad (Khine y Wang 2017; Miloslavskaya y Tolstoy 2016).
Dado que un lago de datos está destinado al almacenamiento y procesamiento de grandes datos, es esencial abordar este problema. Por lo tanto, modificamos la definición de Madera y Laurent para alinearla con nuestra visión e introducir escalabilidad (Sawadogo et al. 2019).

Definición 1 Un lago de datos es un sistema escalable de almacenamiento y análisis de datos de cualquier tipo, retenidos en su formato original y utilizado *principalmente* por especialistas en datos (estadísticos, científicos de datos o analistas) para la extracción de conocimiento. Sus características incluyen:

1. un catálogo de metadatos que hace cumplir la calidad de los datos; 2. políticas y herramientas de gobierno de datos; 3. accesibilidad a varios tipos de usuarios; 4. integración de cualquier tipo de datos; 5. una organización lógica y física; 6. escalabilidad en términos de almacenamiento y procesamiento.

3 Arquitecturas y tecnologías de lagos de datos

Las revisiones existentes sobre arquitecturas de lagos de datos comúnmente distinguen arquitecturas de estanques y zonas (Giebler et al. 2019; Ravat y Zhao 2019a). Sin embargo, esta categorización a veces puede ser confusa. Por lo tanto, presentamos en la Sección 3.1 una nueva forma de clasificar las arquitecturas de lagos de datos que llamamos Funcional × Madurez. Además, presentamos en la Sección 3.2 una lista de posibles tecnologías para implementar un lago de datos. Eventualmente, investigamos en la Sección 3.3 cómo se puede asociar un sistema de lago de datos con un almacén de datos en una arquitectura de datos empresarial.

3.1 Arquitecturas de lagos de datos

3.1.1 Arquitecturas de zona

Arquitectura de estanques Inmon diseña un lago de datos como un conjunto de estanques de datos (Inmon 2016). Un estanque de datos puede verse como una subdivisión de un lago de datos que se ocupa de datos de un tipo específico. De acuerdo con las especificaciones de Dixon, cada estanque de datos está asociado con un sistema de almacenamiento especializado, algún procesamiento y acondicionamiento de datos específicos (es decir, datos

transformación/preparación) y un servicio de análisis pertinente. Más precisamente, Inmon identifica cinco estanques de datos (Fig. 2).

1. El **estanque de datos sin procesar** trata con datos sin procesar recién incorporados. En realidad, es una zona de tránsito, ya que los datos se acondicionan y transfieren a otro estanque de datos, es decir, el estanque de datos analógico, de aplicación o de texto. El estanque de datos sin procesar, a diferencia de los otros estanques, no está asociado con ningún sistema de metadatos.
2. Los datos almacenados en el **estanque de datos analógicos** se caracterizan por una frecuencia muy alta de mediciones, es decir, llegan a alta velocidad. Por lo general, los datos semiestructurados de IoT se procesan en el estanque de datos analógicos.
3. Los datos incorporados en el **estanque de datos de la aplicación** provienen de aplicaciones de software y, por lo tanto, generalmente son datos estructurados de sistemas de gestión de bases de datos relacionales (DBMS). Dichos datos se integran, transforman y preparan para el análisis; e Inmon en realidad considera que el depósito de datos de la aplicación es un almacén de datos.
4. El **estanque de datos textuales** administra datos textuales no estructurados. Cuenta con un texto proceso de desambiguación para facilitar el análisis de datos textuales.
5. El propósito del **depósito de datos de archivo** es guardar los datos que no se utilizan activamente, pero que aún podrían ser necesarios en el futuro. Los datos archivados pueden tener su origen en los estanques de datos analógicos, de aplicación y de texto.

Arquitecturas de zona Las denominadas arquitecturas de zona asignan datos a una zona según su grado de refinamiento (Giebler et al. 2019). Por ejemplo, el lago de datos de Zaloni (LaPlante y Sharma 2016) adopta una arquitectura de seis zonas (Fig. 3).

1. La **zona de carga transitoria se** ocupa de los datos bajo ingesta. Aquí, la calidad básica de los datos se realizan comprobaciones.
2. La **zona de datos sin procesar** maneja datos en formato casi sin procesar provenientes de la zona transitoria.
3. La **zona de confianza** es donde se transfieren los datos una vez estandarizados y limpios.
4. Desde el área de confianza, los datos pasan a la zona de pruebas de **descubrimiento ,** donde los científicos de datos pueden acceder a ellos a través de operaciones de disputa o descubrimiento de datos.
5. Además del sandbox de detección, la **zona de consumo** permite a los usuarios empresariales ejecutar escenarios “qué pasaría si” a través de herramientas de tablero.
6. La **zona de gobierno** finalmente permite administrar, monitorear y gobernar metadatos, calidad de datos, un catálogo de datos y seguridad.

Sin embargo, esta es solo una de varias variantes de arquitecturas de zona. De hecho, tales arquitecturas generalmente difieren en el número y las características de las zonas (Giebler et al. 2019), por ejemplo,

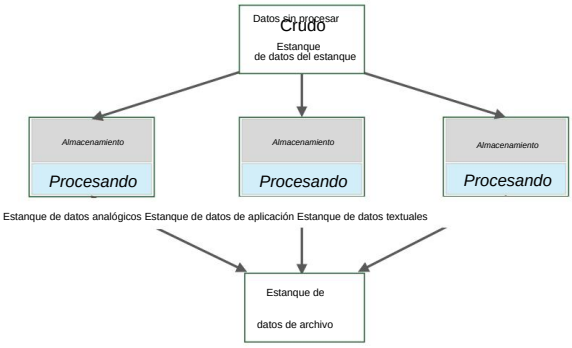


Fig. 2 Flujo de datos en una arquitectura de estanque

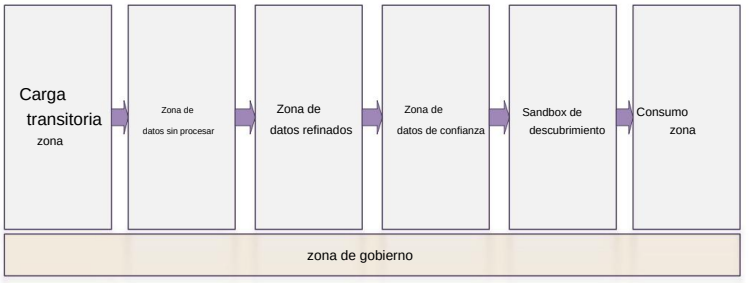


Fig. 3 Arquitectura de zonas de Zaloni (LaPlante y Sharma 2016)

algunas arquitecturas incluyen una zona transitoria (LaPlante y Sharma 2016; Tharrington 2017; Zikopoulos et al. 2015) mientras que otras no (Hai et al. 2016; Ravat y Zhao 2019a).

Una arquitectura de zona particular que se menciona a menudo en la literatura del lago de datos es la arquitectura lambda (John y Misra 2017; Mathis 2017). De hecho, se destaca porque incluye dos zonas de procesamiento de datos: una zona de procesamiento por lotes para datos masivos y una zona de procesamiento en tiempo real para datos rápidos de IoT (John y Misra 2017). Estas dos zonas ayudan a manejar datos rápidos y datos masivos de forma adaptada y especializada.

Discusión Tanto en las arquitecturas de estanques como de zonas, los datos se procesan previamente. Por lo tanto, los análisis son rápidos y fáciles. Sin embargo, esto tiene el costo de la pérdida de datos en las arquitecturas de los estanques, ya que los datos sin procesar se eliminan cuando se transfieren a otros estanques. Los inconvenientes de las muchas arquitecturas de zona dependen de cada variante. Por ejemplo, en la arquitectura de Zaloni (LaPlante y Sharma 2016), los datos fluyen a través de seis áreas, lo que puede generar múltiples copias de los datos y, por lo tanto, dificultades para controlar el linaje de datos. En la arquitectura Lamda (John y Misra 2017), los componentes de velocidad y procesamiento por lotes siguen diferentes paradigmas. Por lo tanto, los científicos de datos deben manejar dos lógicas distintas para los análisis cruzados (Mathis 2017), lo que hace que el análisis de datos sea más difícil en general.

Además, la distinción de arquitecturas de lagos de datos en enfoques de estanques y zonas no es tan nítida en nuestra opinión. De hecho, la arquitectura del estanque puede considerarse como una variante de la arquitectura de zona, ya que la ubicación de los datos depende del nivel de refinamiento de los datos, como en las arquitecturas de zona. Además, algunas arquitecturas de zona incluyen una zona de almacenamiento global donde los datos sin procesar y limpios se almacenan por completo (John y Misra 2017; Quix y Hai 2018), lo que contradice la definición de arquitecturas de zona, es decir, los componentes dependen del grado de refinamiento de los datos.

3.1.2 Arquitecturas funcionales × madurez

Para superar las contradicciones de la categorización estanque/zona, proponemos una forma alternativa de agrupar las arquitecturas de lagos de datos con respecto al tipo de criterios utilizados para definir los componentes. Como resultado, distinguimos arquitecturas funcionales, arquitecturas basadas en la madurez de datos y arquitecturas híbridas (Fig. 4).

Las **arquitecturas funcionales** siguen algunas funciones básicas para definir los componentes de un lago. Las funciones básicas del lago de datos suelen incluir (LaPlante y Sharma 2016):

1. una función de ingestión de datos para conectarse con fuentes de datos; 2.
una función de almacenamiento de datos para conservar datos sin procesar y refinados;

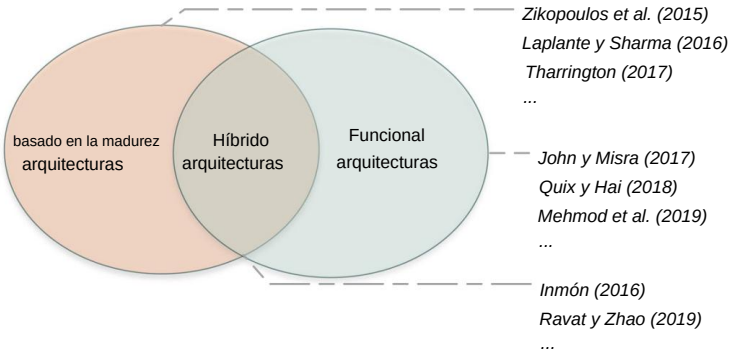


Fig. 4 Propuesta de tipología de arquitectura

3. una función de procesamiento de datos;
4. una función de acceso a datos para permitir la consulta de datos sin procesar y refinados.

Quix y Hai, así como Mehmood et al., basan sus arquitecturas de lagos de datos en estas funciones (Mehmood et al. 2019; Quix y Hai 2018). De manera similar, la arquitectura lambda de John y Misra (John y Misra 2017) puede considerarse una arquitectura funcional, ya que sus componentes representan funciones de lago de datos como almacenamiento, procesamiento y servicio.

Las arquitecturas basadas en la madurez de los datos son arquitecturas de lagos de datos donde los componentes se definen con respecto al nivel de refinamiento de los datos. En otras palabras, está constituido por la mayoría de las arquitecturas zonales. Un buen representante es la arquitectura del lago de datos de Zaloni (LaPlante y Sharma 2016), donde las zonas básicas comunes son una zona transitoria, una zona de datos sin procesar, una zona de datos de confianza y una zona de datos refinados (LaPlante y Sharma 2016; Tharrington 2017; Zikopoulos et al. . 2015).

Las arquitecturas híbridas son arquitecturas de lago de datos en las que los componentes identificados dependen tanto de las funciones del lago de datos como del refinamiento de los datos. La arquitectura del estanque de Inmon es en realidad una arquitectura híbrida (Inmon 2016). Por un lado, es una arquitectura basada en la madurez de los datos, ya que los datos brutos se gestionan en un componente especial, es decir, el estanque de datos brutos, mientras que los datos refinados se gestionan en otros estanques, es decir, los estanques de datos textuales, analógicos y de aplicación. Pero, por otro lado, la arquitectura del estanque también es funcional porque las especificaciones de Inmon consideran algunos componentes de almacenamiento y proceso distribuidos en los estanques de datos (Fig. 2). Ravat y Zhao también proponen una arquitectura de lago de datos híbrido de este tipo (Fig. 5 (Ravat y Zhao 2019a)).

Discusión Las arquitecturas funcionales tienen la ventaja de resaltar claramente las funciones que se implementarán para un lago de datos determinado, lo que ayuda a que coincida fácilmente con las tecnologías requeridas. Por el contrario, las arquitecturas basadas en la madurez de los datos son útiles para planificar y organizar el ciclo de vida de los datos. Por lo tanto, ambos enfoques están limitados, ya que solo se enfocan en un único punto de vista, mientras que, en nuestra opinión, es importante tener en cuenta tanto la funcionalidad como la madurez de los datos al diseñar un lago de datos.

En consecuencia, abogamos por enfoques híbridos. Sin embargo, la arquitectura híbrida existente aún se puede mejorar. Por ejemplo, en el enfoque de estanque de datos de Inmon, los datos sin procesar se eliminan una vez que se refinan. Este proceso puede provocar alguna pérdida de datos, lo que es contrario al espíritu de los lagos de datos. En la propuesta de Ravat y Zhao, el acceso a los datos solo parece posible para datos refinados. Tales limitaciones sugieren que hoy en día todavía se necesita una arquitectura híbrida de lago de datos más completa.

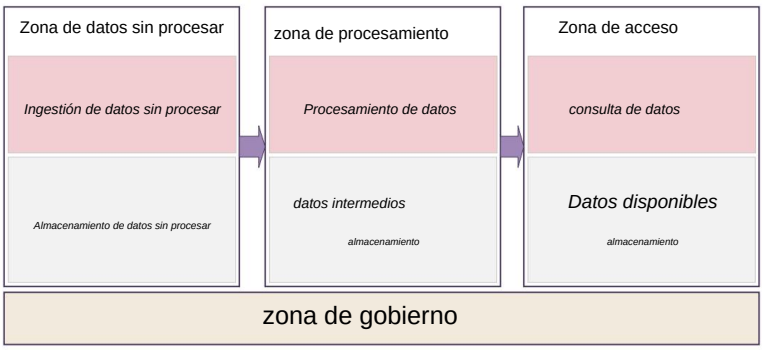


Fig. 5 Arquitectura híbrida de Ravat y Zhao (Ravat y Zhao 2019a)

3.2 Tecnologías para lagos de datos

La mayoría de las implementaciones de lagos de datos se basan en el ecosistema Apache Hadoop (Couto et al. 2019; Khine y Wang 2017). De hecho, Hadoop tiene la ventaja de proporcionar tanto almacenamiento con el sistema de archivos distribuidos de Hadoop (HDFS) como herramientas de procesamiento de datos a través de MapReduce o Spark. Sin embargo, Hadoop no es la única tecnología adecuada para implementar un lago de datos.

En esta sección, vamos más allá de Hadoop para revisar las herramientas utilizables para implementar las funciones básicas del lago de datos.

3.2.1 Ingesta de datos

Las tecnologías de ingestión ayudan a transferir datos físicamente desde fuentes de datos a un lago de datos. Una primera categoría de herramientas incluye software que recopila datos de forma iterativa a través de trabajos prediseñados e industrializados. La mayoría de estas herramientas son propuestas por Apache Foundation y también pueden servir para agregar, convertir y limpiar datos antes de la ingesta. Incluyen Flink y Samza (marcos de procesamiento de flujo distribuido), Flume (un servicio de transferencia de registros de Hadoop), Kafka (un marco que proporciona canalizaciones de datos en tiempo real y aplicaciones de procesamiento de flujo) y Sqoop (un marco para la integración de datos de DBMS SQL y NoSQL en Hadoop)

(John y Misra 2017; Mathis 2017; Suriarachchi y Plale 2016).

Una segunda categoría de tecnologías de ingesta de datos está compuesta por herramientas y protocolos comunes de transferencia de datos (wget, rsync, FTP, HTTP, etc.), que utiliza el administrador del lago de datos dentro de los scripts de ingesta de datos. Tienen la ventaja clave de estar fácilmente disponibles y ser ampliamente comprendidos (Terrizzano et al. 2015). De manera similar, algunas interfaces de programación de aplicaciones (API) están disponibles para la recuperación y transferencia de datos desde la Web a un lago de datos. Por ejemplo, CKAN y Socrata proporcionan API para acceder a un catálogo de datos abiertos y metadatos asociados (Terrizzano et al. 2015).

3.2.2 Almacenamiento de datos

Distinguimos dos enfoques principales para almacenar datos en lagos de datos. La primera forma consiste en utilizar bases de datos clásicas para el almacenamiento. De hecho, algunos lagos de datos usan DBMS relacionales como MySQL, PostgreSQL u Oracle para almacenar datos estructurados (Beheshti et al. 2017; Khine y Wang 2017). Sin embargo, los DBMS relacionales no se adaptan bien a los datos semiestructurados, y más aún a los datos no estructurados. Por lo tanto, los DBMS NoSQL (no solo SQL) generalmente se usan

en cambio (Beheshti et al. 2017; Giebler et al. 2019; Khine y Wang 2017). Además, suponiendo que la variedad de datos sea la norma en los lagos de datos, un sistema de almacenamiento multiparadigma es particularmente relevante (Nogueira et al. 2018). Estos llamados sistemas multitienda administran múltiples DBMS, cada uno de los cuales se adapta a una necesidad de almacenamiento específica.

La segunda forma principal de almacenar datos y la más utilizada es el almacenamiento HDFS (en aproximadamente el 75 % de los lagos de datos (Russom 2017)). HDFS es un sistema de almacenamiento distribuido que ofrece una escalabilidad muy alta y maneja todo tipo de datos (John y Misra 2017). Por lo tanto, es ideal para el almacenamiento masivo y sin esquemas que se necesita para datos no estructurados. Otra ventaja de esta tecnología es la distribución de datos que permite una alta tolerancia a fallos. Sin embargo, HDFS por sí solo no es suficiente para manejar todos los formatos de datos, especialmente los datos estructurados. Por lo tanto, idealmente debería combinarse con DBMS relacionales y/o NoSQL.

3.2.3 Tratamiento de datos

En los lagos de datos, el procesamiento de datos se realiza muy a menudo con MapReduce (Couto et al. 2019; John y Misra 2017; Khine y Wang 2017; Mathis 2017; Stein y Morrison 2014; Suri arachchi y Plale 2016), un paradigma de procesamiento de datos paralelo proporcionado por Apache Hadoop.
MapReduce se adapta bien a datos muy grandes, pero es menos eficiente para datos rápidos porque funciona en disco (Tiao 2018). Así, se utilizan frameworks de procesamiento alternativos, de los cuales el más famoso es Apache Spark. Spark funciona como MapReduce, pero adopta un enfoque de memoria completa en lugar de usar el sistema de archivos para almacenar resultados intermedios. Por lo tanto, Spark es especialmente adecuado para el procesamiento en tiempo real. De manera similar, Apache Flink y Apache Storm también son adecuados para el procesamiento de datos en tiempo real (John y Misra 2017; Khine y Wang 2017; Mathis 2017; Suriarachchi y Plale 2016; Tiao 2018). Sin embargo, estos dos enfoques se pueden implementar simultáneamente en un lago de datos, con MapReduce dedicado a datos voluminosos y motores de procesamiento de flujo para datos veloces (John y Misra 2017; Suriarachchi y Plale 2016).

3.2.4 Acceso a datos

En los lagos de datos, se puede acceder a los datos a través de lenguajes de consulta clásicos como SQL para DBMS relacionales, JSONiq para MongoDB, XQuery para DBMS XML o SPARQL para recursos RDF (Farid et al. 2016; Fauduet y Peyrard 2010; Hai et al. 2016). ; Laskowski 2016; Pathirana 2015). Sin embargo, esto no permite realizar consultas simultáneas en bases de datos heterogéneas, mientras que los lagos de datos almacenan datos heterogéneos y, por lo tanto, normalmente requieren sistemas de almacenamiento heterogéneos.

Una solución a este problema es adoptar las técnicas de consulta de multitiendas (Sección 3.2.2) (Nogueira et al. 2018). Por ejemplo, Spark SQL y SQL++ se pueden usar para consultar DBMS relacionales y datos semiestructurados en formato JSON. Además, el motor de reescritura de consultas escalables (SQRE) maneja bases de datos de gráficos (Hai et al. 2018).
Finalmente, CloudMdsQL también ayuda a consultar simultáneamente múltiples DBMS relacionales y NoSQL (Leclercq y Savonnet 2018). De manera similar, Apache Phoenix se puede usar para convertir automáticamente consultas SQL en un lenguaje de consulta NoSQL, por ejemplo. Apache Drill permite unir datos de múltiples sistemas de almacenamiento (Beheshti et al. 2017). También se puede acceder a los datos almacenados en HDFS mediante Apache Pig (John y Misra 2017).

Eventualmente, los usuarios comerciales, que requieren herramientas interactivas y fáciles de usar para las tareas de visualización y generación de informes de datos, utilizan ampliamente los servicios de tablero como Microsoft Power BI y Tableau sobre lagos de datos (Couto et al. 2019; Russom 2017).

3.3 Combinar lagos de datos y almacenes de datos

Hay en la literatura dos enfoques principales para combinar un lago de datos y un almacén de datos en un sistema de gestión de datos global. El primer enfoque se refiere al uso de un lago de datos como fuente de datos de un almacén de datos (Sección 3.3.1). El segundo considera los almacenes de datos como componentes de los lagos de datos (Sección 3.3.2).

3.3.1 Lago de datos que genera un almacén de datos

Este enfoque tiene como objetivo aprovechar las características específicas tanto de los lagos de datos como de los almacenes de datos. Dado que los lagos de datos permiten un almacenamiento más fácil y económico de una gran cantidad de datos sin procesar, pueden considerarse áreas de preparación o almacenes de datos operativos (ODS) (Fang 2015; Russom 2017), es decir, almacenes de datos intermedios antes de los almacenes de datos que recopilan datos operativos. datos de varias fuentes antes de que se lleve a cabo el proceso ETL.

Con un lago de datos que se abastece de un almacén de datos, posiblemente con datos semiestructurados, los análisis OLAP industrializados son posibles sobre los datos del lago, mientras que los análisis ad-hoc bajo demanda aún son posibles directamente desde el lago de datos (Fig. 6).

3.3.2 Almacén de datos dentro de un lago de datos

Como se detalla en la Sección 3.1.1, Inmon propone una arquitectura basada en una subdivisión de lagos de datos en los llamados estanques de datos (Inmon 2016). Para Inmon, los estanques de datos estructurados provenientes de aplicaciones operativas son, simple y llanamente, almacenes de datos. Por lo tanto, este enfoque actúa sobre una concepción de los lagos de datos como extensiones de los almacenes de datos.

3.3.3 Discusión

Cuando un lago de datos genera un almacén de datos (Sección 3.3.1), existe una separación funcional clara, ya que los almacenes de datos y los lagos de datos se especializan en análisis industrializados y bajo demanda, respectivamente. Sin embargo, esto viene con un problema de silos de datos.

Por el contrario, el síndrome de silos de datos se puede reducir en el enfoque de Inmon (Sección 3.3.2), ya que todos los datos se gestionan y procesan en una plataforma global única. Por lo tanto, diversos datos pueden combinarse fácilmente a través de análisis de referencias cruzadas, lo que sería imposible si los datos se gestionaran por separado. Además, construir un almacén de datos dentro de un lago de datos global puede mejorar el control del ciclo de vida de los datos. Es decir, debería ser más fácil rastrear y, por lo tanto, reproducir los procesos aplicados a los datos que se ingieren en el almacén de datos, a través del sistema de seguimiento del lago de datos.

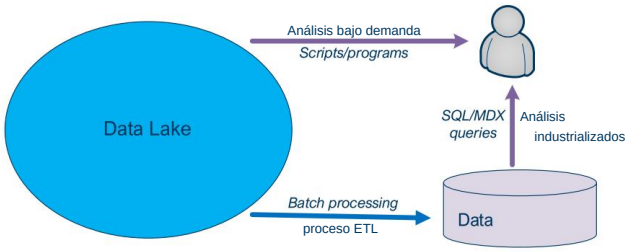


Fig. 6 Arquitectura de lago de datos y almacén de datos

4 Gestión de metadatos en lagos de datos

Los datos incorporados en los lagos de datos no tienen un esquema explícito (Miloslavskaya y Tolstoy 2016), lo que puede convertir fácilmente un lago de datos en un pantano de datos en ausencia de un sistema de metadatos eficiente (Suriarachchi y Plale 2016). Por lo tanto, la gestión de metadatos juega un papel esencial en los lagos de datos (Laskowski 2016; Khine y Wang 2017). En esta sección, detallamos las técnicas de gestión de metadatos utilizadas en los lagos de datos. Primero, identificamos los metadatos que son relevantes para los lagos de datos. Luego, revisamos cómo se pueden organizar los metadatos. También investigamos herramientas y técnicas de extracción de metadatos. Finalmente, proporcionamos un inventario de características deseables en los sistemas de metadatos.

4.1 Categorías de metadatos

Identificamos en la literatura dos tipologías principales de metadatos dedicados a lagos de datos. El primero distingue los metadatos funcionales, mientras que el segundo clasifica los metadatos con respecto a los tipos de metadatos estructurales.

4.1.1 Metadatos funcionales

Oram introduce una clasificación de metadatos en tres categorías, con respecto a la forma en que se recopilan (Oram 2015).

1. **Los metadatos comerciales** se definen como el conjunto de descripciones que hacen que los datos sean más comprensibles y definen las reglas comerciales. Más concretamente, estos suelen ser nombres de campos de datos y restricciones de integridad. Dichos metadatos generalmente los definen los usuarios comerciales en la etapa de ingesta de datos.
2. Los **metadatos operativos** son información generada automáticamente durante el procesamiento de datos. Incluyen descripciones de los datos de origen y de destino, por ejemplo, la ubicación de los datos, el tamaño del archivo, el número de registros, etc., así como la información del proceso.
3. Los **metadatos técnicos** expresan cómo se representan los datos, incluido el formato de datos (p. ej., texto sin formato, imagen JPEG, documento JSON, etc.), estructura o esquema. La estructura de datos consta de características tales como nombres, tipos, longitudes, etc. Se obtienen comúnmente de un DBMS para datos estructurados, o mediante técnicas personalizadas durante la etapa de maduración de datos.

Diamantini et al. mejorar esta tipología con un modelo de metadatos genéricos (Diamantini et al. 2018) y mostrar que los metadatos comerciales, operativos y técnicos a veces se cruzan. Por ejemplo, los campos de datos se relacionan con metadatos comerciales y técnicos, ya que los usuarios comerciales los definen en los esquemas de datos. De manera similar, los formatos de datos pueden considerarse como metadatos tanto técnicos como operativos, y así sucesivamente (Fig. 7).

4.1.2 Metadatos estructurales

En esta clasificación, Sawadogo et al. categorizar los metadatos con respecto a los "objetos" que relacionan con Sawadogo et al. (2019). La noción de objeto puede verse como una generalización del concepto de conjunto de datos (Maccioni y Torlone 2018), es decir, un objeto puede ser una tabla relacional o de hoja de cálculo en un contexto de datos estructurados, o un documento simple (p. ej., documento XML, imagen archivo, archivo de video, documento de texto, etc.) en un contexto de datos semiestructurados o no estructurados. Por lo tanto, usamos el término "objeto" en el resto de este
papel.

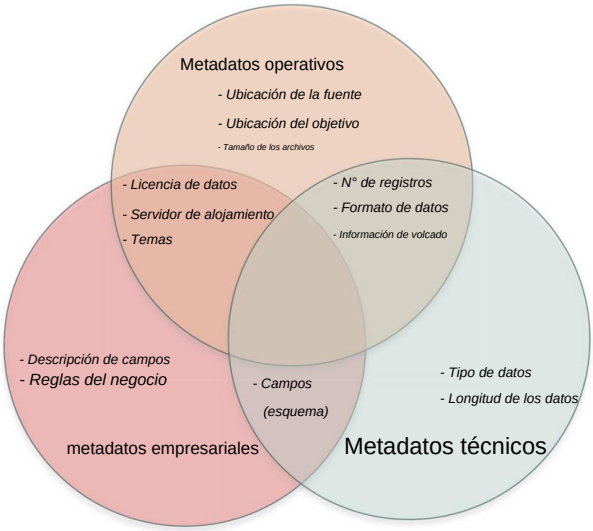


Fig. 7 Modelo de metadatos funcionales (Diamantini et al. 2018)

Los metadatos intraobjeto pertenecen a un conjunto de características asociadas con objetos únicos en el lago. Se subdividen en cuatro subcategorías principales.

1. **Las propiedades** proporcionan la descripción general de un objeto. Por lo general, se recuperan del sistema de archivos como pares clave-valor, por ejemplo, nombre y tamaño del archivo, ubicación, fecha de la última modificación, etc.
2. Los **metadatos de previsualización y resumen** tienen como objetivo proporcionar una visión general del contenido o la estructura de un objeto. Por ejemplo, los metadatos pueden ser esquemas de datos extraídos para datos estructurados y semiestructurados, o nubes de palabras para datos textuales.
3. Los **metadatos de versión y representación** están hechos de datos alterados. Cuando se genera un nuevo objeto de datos o a partir de un objeto existente o en el lago de datos, o se puede considerar como metadatos para o. Los metadatos de versión se obtienen a través de actualizaciones de datos, mientras que los metadatos de representación provienen de operaciones de refinamiento de datos. Por ejemplo, una operación de refinado puede consistir en vectorizar un documento de texto en una bolsa de palabras para su posterior procesamiento automático.
4. **Los metadatos semánticos** implican anotaciones que describen el significado de los datos en un objeto. Incluyen información como título, descripción, categorización, etiquetas descriptivas, etc. A menudo permiten la vinculación de datos. Los metadatos semánticos pueden generarse utilizando recursos semánticos, como ontologías, o agregarse manualmente por usuarios comerciales (Hai et al. 2016; Quix et al. 2016).

Los metadatos entre objetos representan vínculos entre dos o más objetos. Se subdividen en tres categorías.

1. **Las agrupaciones** de objetos organizan los objetos en colecciones. Cualquier objeto puede estar asociado a varias colecciones. Dichos enlaces se pueden deducir automáticamente de algunos metadatos internos del objeto, como etiquetas, formato de datos, idioma, propietario, etc.
2. Los **vínculos de similitud** expresan la fuerza de la similitud entre los objetos. Se obtienen a través de medidas de similitud comunes o personalizadas. Por ejemplo, Maccioni y Torlone (2018) definen las medidas de afinidad y capacidad de unión para expresar la similitud entre objetos semiestructurados.

3. **Los enlaces de paternidad** tienen como objetivo guardar el linaje de datos, es decir, cuando se crea un nuevo objeto a partir del combinación de varios otros, estos metadatos registran el proceso. Los enlaces de paternidad son por lo tanto, se genera automáticamente durante las uniones de datos.

Los metadatos globales son estructuras de datos que proporcionan una capa de contexto para facilitar el procesamiento de datos. y el análisis más fácil. Los metadatos globales no están directamente asociados con ningún objeto específico, pero potencialmente afectar a todo el lago. Hay tres subcategorías de metadatos globales.

1. **Los recursos semánticos** son bases de conocimiento tales como ontologías, taxonomías, tesauros, etc., que ayudan notablemente a mejorar los análisis. Por ejemplo, una ontología se puede utilizar para extender automáticamente una consulta basada en términos con términos equivalentes. Los recursos semánticos son generalmente obtenidos de Internet o construidos manualmente.
2. Los **índices** (incluidos los índices invertidos) mejoran los datos basados en términos o patrones recuperación. Se construyen y enriquecen automáticamente mediante un sistema de indexación.
3. **Los registros** rastrean las interacciones del usuario con el lago de datos, que puede ser simple, por ejemplo, usuario conexión o desconexión, o más compleja, por ejemplo, un trabajo en ejecución.

4.1.3 Discusión

La clasificación de metadatos de Oram es la más citada, especialmente en la literatura industrial (Dia mantini et al. 2018; LaPlante y Sharma 2016; Ravat y Zhao 2019b; Russom 2017), presumiblemente porque está inspirado en las categorías de metadatos de los almacenes de datos (Ravat y Zhao 2019a). Por lo tanto, su adopción parece más fácil y natural para los profesionales que ya están trabajando con él.

Sin embargo, estamos a favor de la segunda clasificación de metadatos, porque incluye la mayoría de las características definidas por Oram. De hecho, los metadatos comerciales son comparables a los metadatos semánticos. Los metadatos operativos pueden considerarse como registros y los metadatos técnicos son equivalentes a metadatos de previsualización. Por lo tanto, la categorización de metadatos estructurales puede considerarse como una extensión, así como una generalización, de la clasificación de metadatos funcionales.

Además, la clasificación de Oram es bastante confusa cuando se aplica en el contexto de lagos de datos. Diamantini et al. de hecho muestran que los metadatos funcionales se cruzan (Sección 4.1.1) (Diamantini et al. 2018). Por lo tanto, los profesionales que no conocen esta tipología pueden confundirse cuando usarlo para identificar y organizar metadatos en un lago de datos.

La Tabla 1 resume los puntos en común y las diferencias entre las dos categorizaciones de metadatos presentadas anteriormente. La comparación aborda el tipo de información tanto proporcionan los inventarios.

Tabla 1 Comparación de Oram y los metadatos de Sawadogo et al. categorías

tipo de información	Funcional metadatos	Estructural metadatos
Características básicas de los datos (tamaño, formato, etc.)		
Semántica de datos (etiquetas, descripciones, etc.)		
Historial de datos Enlace de datos		
Interacciones del usuario		

4.2 Modelado de metadatos

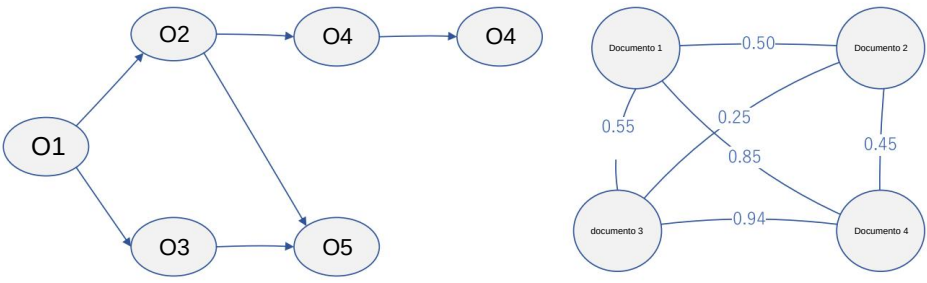
Hay en la literatura dos enfoques principales para representar el sistema de metadatos de un lago de datos.
El primer enfoque, el más común, adopta una vista de gráfico, mientras que el segundo aprovecha la bóveda de datos
modelado.

4.2.1 Modelos de gráficos

La mayoría de los modelos que administran sistemas de metadatos de lagos de datos se basan en un enfoque gráfico. Nosotros
identificar tres subcategorías principales de modelos de metadatos basados en gráficos con respecto a los principales
características a las que apuntan.

Los modelos de gráficos centrados en la procedencia de los datos gestionan principalmente el rastreo de metadatos, es decir, la
información sobre actividades, objetos de datos y usuarios que interactúan con un objeto específico (Suri arachchi y Plale
2016). En otras palabras, rastrean el pedigrí de los objetos de datos (Halevy
et al. 2016). Las representaciones de procedencia generalmente se construyen utilizando un gráfico acíclico dirigido
(DAG) donde los nodos representan entidades como usuarios, roles u objetos (Beheshti et al. 2017;
Hellerstein et al. 2017). Los bordes se utilizan para expresar y describir interacciones entre entidades, por ejemplo, a través de
una marca de tiempo simple, tipo de actividad (leer, crear, modificar) (Beheshti et al.
2017), estado del sistema (CPU, RAM, ancho de banda) (Suriarachchi y Plale 2016) o incluso el
guion utilizado (Hellerstein et al. 2017). Por ejemplo, la Fig. 8a muestra un modelo básico de procedencia
con nodos que representan objetos de datos y bordes que simbolizan operaciones. Procedencia de los datos
el seguimiento ayuda a garantizar la trazabilidad y la repetibilidad de los procesos en los lagos de datos. Por lo tanto,
Los metadatos de procedencia se pueden utilizar para comprender, explicar y reparar incoherencias en el
datos (Beheshti et al. 2017). También pueden servir para proteger datos confidenciales, al detectar
intrusiones (Suriarachchi y Plale 2016).

Los modelos de gráficos centrados en la similitud describen el sistema de metadatos como un gráfico no dirigido
donde los nodos son objetos de datos y los bordes expresan una similitud entre los objetos. tal
la similitud se puede especificar a través de un borde ponderado o no ponderado. Bordes ponderados
mostrar la fuerza de similitud, cuando se utiliza una medida de similitud formal, por ejemplo, afinidad y
unión (Maccioni y Torlone 2018) (Fig. 8b). Por el contrario, los bordes no ponderados sirven para
simplemente detecta si dos objetos están conectados (Farrugia et al. 2016). Tal diseño gráfico
permite realizar análisis de red sobre un lago de datos (Farrugia et al. 2016), por ejemplo, descubrir comunidades o calcular la
centralidad de los nodos y, por lo tanto, su importancia en el lago. otro uso



a) Gráfico de procedencia b) Gráfico de similitud

Fig. 8 Ejemplos de modelos de metadatos basados en gráficos

de similitud de datos puede ser recomendar automáticamente a los usuarios del lago algunos datos relacionados con los datos que observan actualmente (Maccioni y Torlone 2018).

Los modelos de gráficos centrados en la composición ayudan a descomponer cada objeto de datos en varios elementos inherentes. El lago se ve como un DAG donde los nodos representan objetos o atributos, por ejemplo, columnas, etiquetas, etc., y los bordes desde cualquier nodo A hasta cualquier nodo B expresan la restricción B ÿ A (Diamantini et al. 2018; Halevy et al. 2016; Nargesian et al. 2018). Esta organización ayuda a los usuarios a navegar a través de los datos (Nargesian et al. 2018). También se puede utilizar como base para detectar conexiones entre objetos. Por ejemplo, Diamantini et al. (2018) utilizó una medida de cadena simple para detectar vínculos entre objetos heterogéneos comparando sus respectivas etiquetas.

4.2.2 Bóveda de datos

Un lago de datos tiene como objetivo ingerir nuevos datos que posiblemente tengan varias estructuras. Por lo tanto, su sistema de metadatos debe ser flexible para abordar fácilmente nuevos esquemas de datos. Nogueira et al. proponen el uso de una bóveda de datos para abordar este problema (Nogueira et al. 2018). Las bóvedas de datos son, de hecho, modelos lógicos alternativos a los esquemas en estrella de los almacenes de datos que, a diferencia de los esquemas en estrella, permiten una fácil evolución del esquema (Linstedt 2011). El modelado de la bóveda de datos involucra tres tipos de entidades (Hultgren 2016).

1. Un **hub** representa un concepto comercial, por ejemplo, cliente, proveedor, venta o producto en un sistema de decisiones comerciales.
2. Un **enlace** representa una relación entre dos o más concentradores.
3. **Los satélites** contienen información descriptiva asociada con un concentrador o un enlace. Cada satélite está conectado a un concentrador o enlace único. Por el contrario, los enlaces o concentradores pueden estar asociados con cualquier número de satélites.

En la propuesta de Nogueira et al., los metadatos comunes a todos los objetos, por ejemplo, título, categoría, fecha y ubicación, se almacenan en hubs; mientras que los metadatos específicos de algunos objetos únicamente, por ejemplo, el idioma de los documentos de texto o la editorial de los libros, se almacenan en satélites (Fig. 9).
Además, cualquier nuevo tipo de objeto tendría sus metadatos específicos almacenados en un nuevo satélite.

4.2.3 Discusión

El modelado de bóveda de datos rara vez se asocia con lagos de datos en la literatura, presumiblemente porque se asocia principalmente con almacenes de datos. Sin embargo, este enfoque garantiza la evolución del esquema de metadatos, lo cual es necesario para construir un lago de datos eficiente. Otra ventaja del modelado de bóveda de datos es que, a diferencia de los modelos gráficos, se puede implementar de forma intuitiva en un DBMS relacional. Sin embargo, todavía se necesitan varias adaptaciones para que este modelo se ocupe de la vinculación de datos como en los modelos gráficos.

Los modelos de gráficos, aunque requieren sistemas de almacenamiento más específicos, como RDF o DBMS de gráficos, siguen siendo ventajosos porque permiten enriquecer automáticamente el lago con información que facilita y mejora los análisis futuros. Sin embargo, las tres subcategorías de modelos gráficos deben integrarse juntas para este propósito. Este sigue siendo un problema abierto porque, como máximo, dos de estos enfoques de gráficos se implementan simultáneamente en los sistemas de metadatos de la literatura (Diamantini et al. 2018; Halevy et al. 2016). El modelo de metadatos MEDAL incluye las tres subcategorías de modelos gráficos (Sawadogo et al. 2019), pero aún no está implementado.

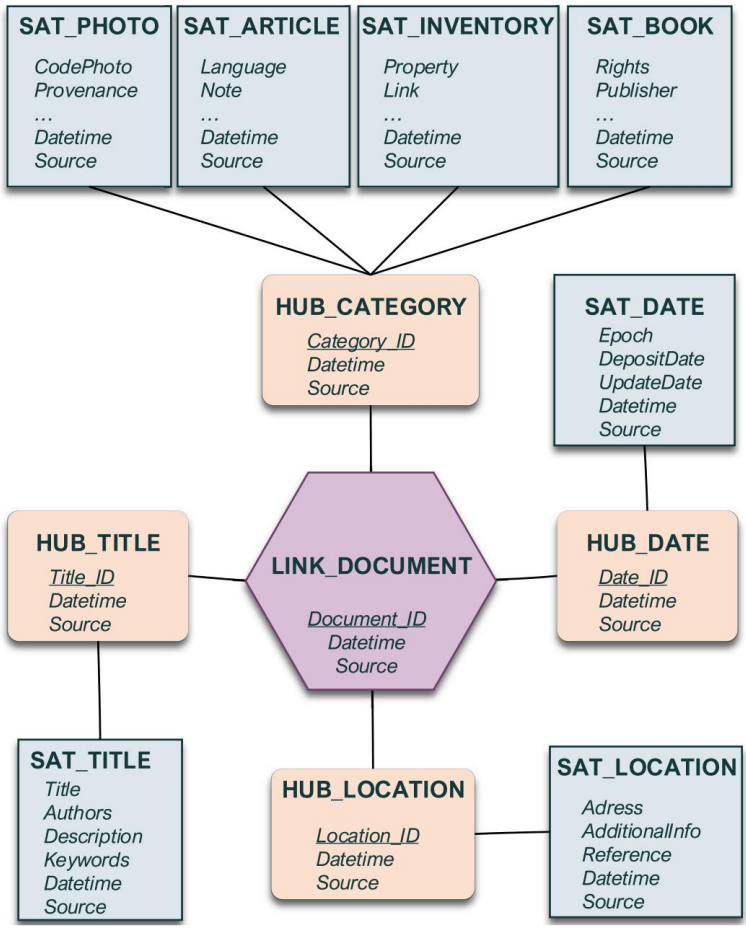


Fig. 9 Modelo de bóveda de metadatos de muestra (Nogueira et al. 2018)

4.3 Generación de metadatos

La mayoría de las herramientas de ingestión de datos de la Sección 3.2.1 también pueden servir para extraer metadatos. Por ejemplo, Suriarachchi y Plale usan Apache Flume para recuperar la procedencia de los datos en un lago de datos (Suriarachchi y Plale 2016). Del mismo modo, las propiedades y la semántica Los metadatos se pueden obtener a través de protocolos especializados como el Comprehensive Knowledge Archive Network (CKAN), un sistema abierto de gestión de almacenamiento de datos (Terriz zano et al. 2015).

Un segundo tipo de tecnologías es más específico para la generación de metadatos. Por ejemplo, Apache Tika ayuda a detectar el tipo MIME y el idioma de los objetos (Quix et al. 2016). Otro herramientas como Open Calais y Alchemy API de IBM también pueden enriquecer los datos a través de identificación de entidades, inferencia de relaciones y detección de eventos (Farid et al. 2016).

Los algoritmos ad-hoc también pueden generar metadatos. Por ejemplo, Singh et al. muestra esa Los modelos bayesianos permiten detectar vínculos entre atributos de datos (Singh et al. 2016). Similarmente,

varios autores proponen algoritmos para descubrir esquemas o restricciones en datos semiestructurados (Beheshti et al. 2017; Klettke et al. 2017; Quix et al. 2016).

Por último, pero no menos importante, Apache Atlas (The Apache Software Foundation 2019), un marco de metadatos ampliamente utilizado (Russom 2017), presenta métodos avanzados de generación de metadatos a través de los llamados ganchos, que son scripts nativos o personalizados que se basan en registros para generar metadatos. Los ganchos ayudan notablemente a Atlas a extraer automáticamente metadatos de linaje y propagar etiquetas en todas las derivaciones de datos etiquetados.

4.4 Características de los sistemas de metadatos de lagos de datos

Un lago de datos que se vuelve inoperable debido a la falta de una gestión adecuada de los metadatos se denomina pantano de datos (Khine y Wang 2017), volcado de datos (Inmon 2016; Suriarachchi y Plale 2016) o lago de datos unidireccional (Inmon 2016), siendo el pantano de datos el términos más comunes. En un lago de datos tan defectuoso, los datos se ingieren, pero nunca se pueden extraer. Por lo tanto, un pantano de datos no puede garantizar ningún análisis. Sin embargo, hasta donde sabemos, no existe una forma objetiva de medir o comparar la eficiencia de los sistemas de metadatos del lago de datos. Por lo tanto, primero presentamos en esta sección una lista de características esperadas para un sistema de metadatos. Luego, presentamos una comparación de dieciocho sistemas de metadatos de lagos de datos con respecto a estas características.

4.4.1 Identificación de características

Sawadogo et al. identifique seis características que un sistema de metadatos de lago de datos debería implementar idealmente para que se considere integral (Sawadogo et al. 2019).

1. **El enriquecimiento semántico (SE)** también se conoce como anotación semántica (Hai et al. 2016) o perfilado semántico (Ansari et al. 2018). Implica agregar información como título, etiquetas, descripción y más para que los datos sean comprensibles (Terrizzano et al. 2015).
   Esto se hace comúnmente utilizando bases de conocimiento como ontologías (Ansari et al. 2018).
   La anotación semántica juega un papel vital en los lagos de datos, ya que hace que los datos sean significativos al proporcionar resúmenes informativos (Ansari et al. 2018). Además, los metadatos semánticos podrían ser la base de la generación de enlaces entre datos (Quix et al. 2016). Por ejemplo, los objetos de datos con las mismas etiquetas podrían considerarse vinculados.
2. **La indexación de datos (DI)** se usa comúnmente en los dominios de base de datos y recuperación de información para encontrar rápidamente un objeto de datos. La indexación de datos se realiza mediante la creación y el enriquecimiento de una estructura de datos que permite la recuperación eficiente de datos del lago. La indexación puede servir tanto para la recuperación simple basada en palabras clave como para consultas más complejas mediante patrones. Todos los datos, ya sean estructurados, semiestructurados o no estructurados, se benefician de la indexación (Singh et al. 2016).
3. **Link Generation (LG)** consiste en identificar e integrar enlaces entre los datos del lago.
   Esto se puede hacer incorporando enlaces preexistentes de fuentes de datos o detectando nuevos enlaces. La generación de enlaces permite análisis adicionales. Por ejemplo, los enlaces de similitud pueden servir para recomendar a los usuarios del lago datos cercanos a los datos que utilizan actualmente (Maccioni y Torlone 2018).
   En la misma línea, los enlaces de datos se pueden utilizar para detectar automáticamente grupos de datos fuertemente vinculados (Farrugia et al. 2016).
4. **El polimorfismo de datos (DP)** es la gestión simultánea de varias representaciones de datos en el lago. Una representación de datos de, por ejemplo, un documento de texto, puede ser una nube de etiquetas o un vector de frecuencias de términos. Los datos semiestructurados y no estructurados deben transformarse al menos parcialmente para que se procesen automáticamente (Diamantini et al. 2018).
   Por lo tanto, el polimorfismo de datos es relevante ya que permite almacenar y reutilizar datos transformados.

Esto hace que los análisis sean más fáciles y rápidos al evitar la repetición de ciertos procesos. (Stefanowski et al. 2017).

5. **El control de versiones de datos (DV)** expresa la capacidad de un sistema de metadatos para administrar las operaciones de actualización, al tiempo que conserva los estados de datos anteriores. Es muy relevante para los lagos de datos, ya que asegura la reproducibilidad del proceso y la detección y corrección de inconsistencias (Bhattacherjee y Deshpande 2018). Además, el control de versiones de datos permite ramificar y evolución de datos concurrentes (Hellerstein et al. 2017).
6. **Usage Tracking (UT)** consiste en gestionar información sobre las interacciones de los usuarios con el lago Tales interacciones son comúnmente operaciones de creación, lectura y actualización. Esto permite seguir de forma transparente la evolución de los objetos de datos. Además, el uso El seguimiento puede servir para la seguridad de los datos, ya sea explicando las inconsistencias de los datos o a través de la detección de intrusos. El seguimiento de uso y el control de versiones de datos están relacionados, ya que las interacciones de actualización a menudo inducen nuevas versiones de datos. Sin embargo, son distintos características, ya que se pueden implementar de forma independiente (Beheshti et al. 2017; Suriarachchi y Plale 2016).

4.4.2 Comparación del sistema de metadatos

Presentamos en la Tabla 2 una comparación de dieciocho sistemas de metadatos de última generación y modelos con respecto a las características que implementan (Sawadogo et al. 2019). Distinguimos los modelos de metadatos de las implementaciones. Los modelos son de hecho bastante teóricos y describir la organización conceptual de los metadatos. Por el contrario, las implementaciones siguen un

Tabla 2 Comparación de sistemas de metadatos de lagos de datos (Sawadogo et al. 2019)

Sistema	Tipo	SE	DI	LG	DP	VD	Utah
SPAR (Fauduet y Peyrard 2010)							
Alrehamy y Walker (2015)							
Terrizzano et al. (2015)							
Constanza (Hai et al. 2016)							
GEMM (Quix et al. 2016)	ÿ						
ALMEJAS (Farid et al. 2016)							
Suriarachchi y Plale (2016)	ÿ						
Singh et al. (2016)							
Farrugia et al. (2016)							
BIENES (Halevy et al. 2016)							
Core DB (Beheshti et al. 2017)							
Suelo (Hellerstein et al. 2017)	ÿ						
KAYAK (Maccioni y Torlone 2018)							
CoreKG (Beheshti et al. 2018)							
Diamantini et al. (2018)	ÿ						
Mehmod et al. (2019)							
CODAL (Sawadogo et al. 2019)							
MEDALLA (Sawadogo et al. 2019)	ÿ						

: Implementaciones ÿ : Modelos de metadatos

: modelo o implementación similar a un lago de datos

enfoque más operativo, pero suelen ser poco detallados, centrándose principalmente en una descripción del sistema resultante en lugar de la metodología aplicada. Esta comparación también considera sistemas de metadatos que no están explícitamente asociados al concepto de lago de datos por parte de sus autores, pero cuyas características permiten ser considerados como tales, por ejemplo, el modelo de metadatos Ground (Hellerstein et al. 2017).

La comparación muestra que el sistema de metadatos más completo con respecto a las funciones que proponemos es MEDAL, con todas las funciones cubiertas. Sin embargo, aún no está implementado. Los siguientes mejores sistemas son GOODS y CoreKG, con cinco de las seis funciones implementadas. Sin embargo, son sistemas de metadatos de caja negra, con pocos detalles sobre la organización conceptual de los metadatos. Por lo tanto, se puede preferir el modelo de metadatos Ground, ya que es mucho más detallado y casi tan completo (cuatro de seis características).

Eventualmente, dos de las seis características definidas en la Sección 4.4.1 pueden considerarse avanzadas. De hecho, el polimorfismo de datos y el control de versiones de datos se encuentran principalmente en los sistemas más completos, como GOODS, CoreKG y Ground. Por lo tanto, su ausencia en la mayoría de los sistemas de metadatos puede atribuirse a la complejidad de la implementación.

5 ventajas y desventajas de los lagos de datos

En esta sección, explicamos los beneficios de usar un lago de datos en lugar de los sistemas de administración de datos más tradicionales, pero también identificamos las dificultades que pueden corresponder a estos beneficios esperados.

Una característica motivadora importante en los lagos de datos es **el almacenamiento económico.** Los lagos de datos son de diez a cien veces menos costosos de implementar que las bases de datos tradicionales orientadas a la toma de decisiones. Esto se puede atribuir al uso de tecnologías de código abierto como HDFS (Khine y Wang 2017; Stein y Morrison 2014). Otra razón es que el almacenamiento en la nube que se usa a menudo para construir lagos de datos reduce el costo de las tecnologías de almacenamiento. Es decir, el propietario del lago de datos paga solo por los recursos realmente utilizados. Sin embargo, el uso de HDFS aún puede alimentar **conceptos erróneos,** ya que el concepto de lago de datos sigue siendo ambiguo para muchos usuarios potenciales. De hecho, a menudo se considera como un sinónimo o una etiqueta de marketing estrechamente relacionada con la tecnología HDFS (Alrehamy y Walker 2015; Grosser et al. 2016).

Otra característica que se encuentra en el centro del concepto de lago de datos es **la fidelidad de los datos.** A diferencia de las bases de datos tradicionales orientadas a la toma de decisiones, los datos originales se conservan en un lago de datos para evitar cualquier pérdida de datos que pudiera ocurrir por las operaciones de preprocesamiento y transformación de datos (Ganore 2015; Stein y Morrison 2014). Sin embargo, la fidelidad de los datos induce un alto riesgo de **inconsistencia** de datos en los lagos de datos, debido a la integración de datos de fuentes múltiples y dispares sin ninguna transformación (O'Leary 2014).

Uno de los principales beneficios de los lagos de datos es que permiten explotar y analizar **datos no estructurados** (Ganore 2015; Laskowski 2016; Stein y Morrison 2014). Esta es una ventaja significativa cuando se trata de grandes datos, que son predominantemente no estructurados (Miloslavskaya y Tolstoy 2016). Además, debido al enfoque de lectura de esquema, los lagos de datos pueden cumplir con cualquier tipo y formato de datos (Cha et al. 2018; Ganore 2015; Khine y Wang 2017; Madera y Laurent 2016). Por lo tanto, los lagos de datos permiten una gama más amplia de análisis que las bases de datos tradicionales orientadas a la toma de decisiones, es decir, almacenes de datos y datamarts, y por lo tanto muestran una mayor **flexibilidad y agilidad.** Sin embargo, aunque el concepto de lago de datos data de 2010, recién se ha puesto en práctica a mediados de la década de 2010. Así, las implementaciones varían, aún están madurando y **faltan estándares metodológicos y técnicos,** lo que sustenta confusiones sobre los lagos de datos. Finalmente, debido a la ausencia de un esquema explícito, **los servicios de acceso a datos** y las API son esenciales para permitir la extracción de conocimiento.

en un lago de datos. En otras palabras, un servicio de acceso a datos es imprescindible para construir con éxito un lago de datos (Alrehamy y Walker 2015; Inmon 2016), aunque dicho servicio no siempre está presente.

A continuación, una ventaja aclamada de los lagos de datos sobre los almacenes de datos es **la ingesta de datos en tiempo real.** De hecho, los datos se incorporan en lagos de datos sin ninguna transformación, lo que evita cualquier lapso de tiempo entre la extracción de datos de las fuentes y su incorporación en el lago de datos (Ganore 2015; Laskowski 2016). Pero como consecuencia, un lago de datos requiere un **sistema de metadatos** eficiente para garantizar el acceso a los datos. Sin embargo, el problema radica en el "cómo", es decir, el uso de métodos o tecnologías inapropiados para construir el sistema de metadatos puede convertir fácilmente el lago de datos en un pantano de datos inoperable (Alrehamy y Walker 2015).

Desde un punto de vista más técnico, los lagos de datos y los análisis relacionados suelen implementarse utilizando tecnologías distribuidas, por ejemplo, HDFS, MapReduce, Apache Spark, Elasticsearch, etc. Estas tecnologías suelen proporcionar una **alta escalabilidad** (Fang 2015; Miloslavskaya y Tolstoy 2016). Además, la mayoría de las tecnologías utilizadas en los lagos de datos tienen mecanismos de replicación, por ejemplo, Elasticsearch, HDFS, etc. Dichas tecnologías permiten una alta resistencia a fallas tanto de hardware como de software y hacen cumplir la **tolerancia a fallas** (John y Misra 2017).

Eventualmente, los lagos de datos a menudo se ven como cajas de arena donde los analistas pueden "jugar", es decir, acceder y preparar datos para realizar varios **análisis específicos sobre la marcha** (Russom 2017; Stein y Morrison 2014). Sin embargo, tal escenario requiere **experiencia.** De hecho, los usuarios de lagos de datos suelen ser científicos de datos (Khine y Wang 2017; Madera y Laurent 2016), lo que contrasta con los sistemas de decisión tradicionales, donde los usuarios comerciales pueden operar el sistema. Por lo tanto, un lago de datos induce una mayor necesidad de perfiles específicos y, por lo tanto, más costosos. De hecho, los científicos de datos deben dominar un amplio conocimiento y una panoplia de tecnologías.

Además, con la integración en lagos de datos de datos estructurados, semiestructurados y no estructurados, los científicos de datos expertos pueden descubrir vínculos y **correlaciones entre datos heterogéneos** (Ganore 2015). Los lagos de datos también permiten integrar fácilmente datos "tal cual" de fuentes externas, por ejemplo, la Web o las redes sociales. Dichos datos externos pueden luego asociarse con datos patentados para generar nuevos conocimientos a través **de análisis cruzados** (Laskowski 2016). Sin embargo, varios enfoques estadísticos y de inteligencia artificial (IA) no están diseñados originalmente para operaciones paralelas, ni para transmisión de datos, por ejemplo, K-medias o K-vecinos más cercanos. Por lo tanto, es necesario **reajustar los enfoques estadísticos y de IA clásicos** para que coincidan con los entornos distribuidos que se utilizan a menudo en los lagos de datos (O'Leary 2014), lo que a veces resulta difícil.

6. Conclusión

En este documento de encuesta, establecemos un estado del arte integral de los diferentes enfoques para el diseño y construimos conceptualmente un lago de datos. En primer lugar, enunciamos las definiciones del concepto de lago de datos y complementamos el mejor existente. En segundo lugar, investigamos arquitecturas y tecnologías alternativas para lagos de datos y proponemos una nueva tipología de arquitecturas de lagos de datos. En tercer lugar, revisamos y analizamos las técnicas de gestión de metadatos utilizadas en los lagos de datos. En particular, clasificamos los metadatos e introducimos las características necesarias para lograr un sistema completo de metadatos. Cuarto, discutimos los pros y los contras de los lagos de datos. En quinto lugar, resumimos mediante un mapa mental los conceptos clave presentados en este documento (Fig. 10).

Finalmente, en eco de los temas que elegimos *no* abordar en este documento (Sección 1), nos gustaría abrir la discusión sobre importantes temas de investigación actuales en el campo de los lagos de datos.

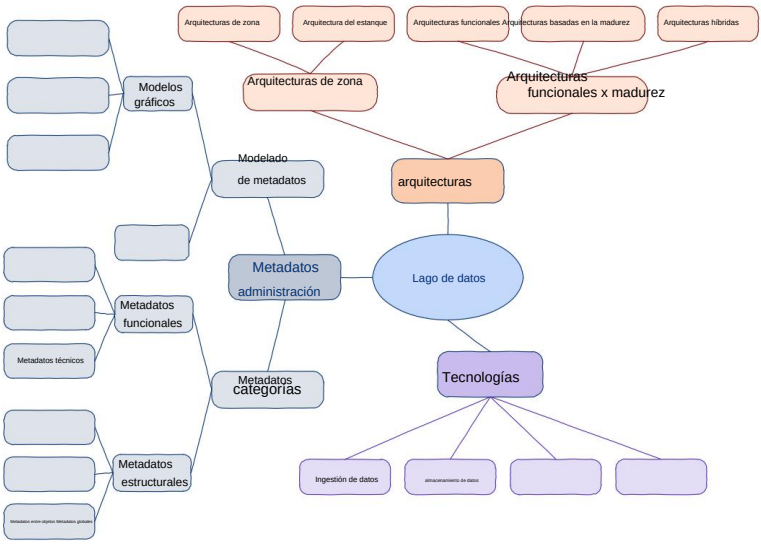


Fig. 10 Conceptos clave investigados en esta encuesta

La integración y transformación de datos han sido durante mucho tiempo temas recurrentes. Aunque retrasados, todavía están presentes en los lagos de datos y se vuelven aún más desafiantes por el volumen, la variedad, la velocidad y la falta de veracidad de los grandes datos. Además, al transformar dichos datos, las funciones definidas por el usuario (UDF) deben usarse con mucha frecuencia (tareas MapReduce, por lo general). En los procesos ETL y ELT, las UDF son mucho más difíciles de optimizar que las consultas clásicas, un problema que aún no se aborda en la literatura (Stefanowski et al. 2017).

Dado que las soluciones de almacenamiento de datos actualmente van más allá de HDFS en lagos de **datos, la interrogación** de datos a través de metadatos sigue siendo un desafío. De hecho, los almacenes múltiples y los almacenes múltiples proporcionan soluciones unificadas para datos estructurados y semiestructurados, pero no abordan los datos no estructurados, que actualmente se consultan por separado a través de almacenes de índice. Además, cuando se considera la gravedad de los datos (Madera y Laurent 2016), la integración de datos virtuales se convierte en una solución relevante. Sin embargo, es probable que los enfoques de mediación requieran nuevos enfoques de almacenamiento en caché y optimización de consultas adaptados a big data (Quix y Hai 2018; Stefanowski et al. 2017).

Los conjuntos de **datos no estructurados ,** aunque unánimemente reconocidos como ubicuos y fuentes de información crucial, se abordan muy poco de manera específica en la literatura relacionada con los lagos de datos. Por lo general, se mencionan el almacenamiento de índices y la minería de texto, pero no hay una reflexión profunda sobre soluciones de análisis o consultas globales. Además, la explotación de otros tipos de datos no estructurados que no sean texto, por ejemplo, imágenes, sonidos y videos, ni siquiera está prevista a día de hoy.

Nuevamente, aunque todos los actores en el dominio del lago de datos enfatizan la importancia de la gobernanza de **datos** para evitar que un lago de datos se convierta en un pantano de datos, la calidad de los datos, la seguridad, la gestión del ciclo de vida y el linaje de los metadatos se consideran riesgos en lugar de problemas que deben abordarse *a priori* . en lagos de datos (Madera y Laurent 2016). De hecho, los principios de gobierno de datos actualmente rara vez se convierten en soluciones reales.

Finalmente, **la seguridad de los datos** actualmente se aborda desde un punto de vista técnico en los lagos de datos, es decir, a través del control de acceso y privilegios, el aislamiento de la red, por ejemplo, con herramientas Docker (Cha et al. 2018), el cifrado de datos y los motores de búsqueda seguros (Maroto 2018).). Sin embargo,

Más allá de estos problemas y los que ya se abordan en el gobierno de datos (integridad, consistencia, disponibilidad) y/o relacionados con el Reglamento General de Protección de Datos (GDPR) europeo, al almacenar y analizar de forma cruzada grandes volúmenes de diversos datos, los lagos de datos permiten mashups que potencialmente inducir graves violaciones de la privacidad de los datos (Joss 2016). Estos temas todavía se investigan a día de hoy.

Agradecimientos La investigación explicada en este documento fue financiada por la Universite Lumiere Lyon 2 y la región de Auvergne-Rhone-Alpes a través de los proyectos COREL y AURA-PMI, respectivamente. Los autores también agradecen sinceramente a los revisores anónimos de este documento por sus comentarios y sugerencias constructivas.

Referencias

Alrehamy, H. y Walker, C. (2015). Lago de datos personales con atracción de gravedad de datos. En *la quinta conferencia internacional de IEEE sobre big data y computación en la nube (BDCloud 2015), Dalian, China, IEEE computer society washington, vol. 88, págs. 160–167.* https://doi.org/10.1109/BDCloud.2015.62.

Ansari, JW, Karim, N., Decker, S., Cochez, M., Beyan, O. (2018). Ampliación de la gestión de metadatos del lago de datos mediante perfiles semánticos. En *2018 Conferencia web semántica extendida (ESWC 2018), Heraklion, Creta, Grecia, págs. 1–15.*

Beheshti, A., Benatallah, B., Nouri, R., Chhieng, VM, Xiong, H., Zhao, X. (2017). CoreDB: un servicio de lago de datos. En *2017 ACM On conference on information and Knowledge Management (CIKM 2017), Singapur, Singapur, ACM, págs. 2451–2454.* https://doi.org/10.1145/3132847.3133171.

Beheshti, A., Benatallah, B., Nouri, R., Tabebordbar, A. (2018). CoreKG: un servicio de lago de conocimiento. *Actas de la Fundación VLDB,* 11(12), 1942–1945. https://doi.org/10.14778/3229863.3236230.

Bhattacherjee, S. y Deshpande, A. (2018). RSTore: un almacén de documentos de varias versiones distribuido. En *IEEE 34th conferencia internacional sobre ingeniería de datos (ICDE), París, Francia, págs. 389–400.* https://doi.org/10.1109/ICDE.2018.00043.

Cha, B., Park, S., Kim, J., Pan, S., Shin, J. (2018). Pruebas de rendimiento y seguridad de la red internacional basadas en un clúster de almacenamiento de abismo distribuido y un borrador del marco del lago de datos. *Redes de comunicación y seguridad de Hindawi, 2018,* 1–14. https://doi.org/10.1155/2018/1746809.

Chessell, M., Scheepers, F., Nguyen, N., van Kessel, R., van der Starre, R. (2014). Gobernando y administrando big data para análisis y tomadores de decisiones. IBM.

Couto, J., Borges, O., Ruiz, D., Marczak, S., Prikladnicki, R. (2019). Un estudio de mapeo sobre lagos de datos: una definición mejorada y posibles arquitecturas. En *31ª conferencia internacional sobre ingeniería de software e ingeniería del conocimiento (SEKE 2019), Lisboa, Portugal, pp. 453–458.* https://doi.org/10.18293/SEKE2019-129.

Diamantini, C., Giudice, PL, Musarella, L., Potena, D., Storti, E., Ursino, D. (2018). Un nuevo modelo de metadatos para manejar de manera uniforme fuentes heterogéneas de lagos de datos. En: Nuevas tendencias en bases de datos y sistemas de información - ADBIS 2018 Short Papers and Workshop, Budapest, Hungría, pp. 165–177. https://doi.org/10.1007/978-3-030-00063-9 17.

Dixon, J. (2010). Pentaho, Hadoop y lagos de datos. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-y-lagos-de-datos/.

Colmillo, H. (2015). Gestión de lagos de datos en la era de Big Data: qué es un lago de datos y por qué se ha vuelto popular en el ecosistema de gestión de datos. En *la quinta conferencia internacional anual de IEEE sobre tecnología cibernética en automatización, control y sistemas inteligentes (CYBER 2015), Shenyang, China, IEEE, págs. 820–824.* https://doi.org/10.1109/CYBER.2015.7288049.

Farid, M., Roatis, A., Ilyas, IF, Hoffmann, HF, Chu, X. (2016). CLAMS: aportando calidad a los lagos de datos. En *2016 Conferencia internacional sobre gestión de datos (SIGMOD 2016), San Francisco, CA, EE. UU., ACM, pp. 2089–2092.* https://doi.org/10.1145/2882903.2899391.

Farrugia, A., Claxton, R., Thompson, S. (2016). Hacia el análisis de redes sociales para comprender y administrar lagos de datos empresariales. En *Avances en análisis y minería de redes sociales (ASONAM 2016), San Francisco, CA, EE. UU., IEEE, págs. 1213–1220.* https://doi.org/10.1109/ASONAM.2016.7752393.

Fauduet, L. y Peyrard, S. (2010). Una estrategia de preservación de datos primero: gestión de datos en spar. En: 7ª conferencia internacional sobre preservación de objetos digitales (iPRES 2010), Viena, Autria, pp. 1–8. http://www.ifs.tuwien.ac.at/dp/ipres2010/papers/fauduet-13.pdf .

Ganore, P. (2015). Introducción al concepto de Data Lake y sus beneficios. https://www.esds.co.in/blog/introducción-al-concepto-de-data-lake-y-sus-beneficios.

Giebler, C., Groger, C., Hoos, E., Schwarz, H., Mitschang, B. (2019). Aprovechamiento del lago de datos: estado actual y desafíos. En *Actas de la 21.ª conferencia internacional sobre análisis de big data y descubrimiento de conocimiento (DaWaK* (p. 2019). Austria: Linz.

Grosser, T., Bloeme, J., Mack, M., Vitsenko, J. (2016). Hadoop y lagos de datos: casos de uso, beneficios y limitaciones centro de investigación de aplicaciones comerciales - BARC GmbH.

Hai, R., Geisler, S., Quix, C. (2016). Constance: un sistema de lago de datos inteligente. En *Conferencia internacional sobre gestión de datos (SIGMOD 2016), San Francisco, CA, EE. UU., ACM Digital Library, pp. 2097– 2100.* https://doi.org/10.1145/2882903.2899389.

Hai, R., Quix, C., Zhou, C. (2018). Reescritura de consultas para lagos de datos heterogéneos. En: 22.ª conferencia europea sobre avances en bases de datos y sistemas de información (ADBIS 2018), Budapest, Hungría, LNCS, vol. 11019, págs. 35–49. Saltador. https://doi.org/10.1007/978-3-319-98398-1 3.

Halevy, AY, Korn, F., Noy, NF, Olston, C., Polyzotis, N., Roy, S., Whang, SE (2016). Bienes: organizar los conjuntos de datos de Google. En *Actas de la conferencia internacional sobre gestión de datos de 2016 (SIGMOD 2016), San Francisco, CA, EE. UU., págs. 795–806.* https://doi.org/10.1145/2882903.2903730.

Hellerstein, JM, Sreekanti, V., sGonzalez, JE, Dalton, J., Dey, A., Nag, S., Ramachandran, K., Arora, S., Bhattacharyya, A., Das, S., Donsky, M., Fierro, G., Ella, C., Steinbach, C., Subramanian, V., Sun, E. (2017). Terreno: un servicio de contexto de datos. En: 8.ª Conferencia Bienal sobre Investigación de Sistemas de Datos Innovadores (CIDR 2017), Chaminade, CA, EE. UU. http://cidrdb.org/cidr2017/papers/p111-hellerstein-cidr17.pdf .

Hultgren, H. (2016). Guía de modelado de Data Vault: guía de introducción al modelado de Data Vault. Genessee Academia, Estados Unidos.

Inmón, B. (2016). Arquitectura del lago de datos: diseñar el lago de datos y evitar el basurero. Técnica Publicaciones.

John, T. y Misra, P. (2017). Data Lake para empresas: arquitectura Lambda para crear datos empresariales sistemas Publicación de paquetes.

Joss, A. (2016). El auge del lago de datos GDPR. https://blogs.informatica.com/2016/06/16/rise-gdpr-data-lake/.

Khine, PP y Wang, ZS (2017). Lago de datos: una nueva ideología en la era de los grandes datos. En *la cuarta conferencia internacional sobre comunicación inalámbrica y red de sensores (WCSN 2017), Wuhan, China, web de conferencias de ITM, vol. 17, págs. 1 a 6.* https://doi.org/10.1051/itmconf/2018170302.

Klettke, M., Awolin, H., St¨url, U., M¨uller, D., Scherzinger, S. (2017). Descubriendo la historia de la evolución de los lagos de datos. En *la conferencia internacional IEEE de 2017 sobre big data (BIGDATA 2017), Boston, MA, EE. UU., págs. 2462–2471.* https://doi.org/10.1109/BigData.2017.8258204.

LaPlante, A. y Sharma, B. (2016). Arquitectura de lagos de datos arquitecturas de gestión de datos para avanzados casos de uso comercial. O'Reilly Media Inc.

Laskowski, N. (2016). Gobernanza del lago de datos: un gran dato de vida o muerte. https://searchcio.techtarget.com/feature/ Data-lake-governance-A-big-data-do-or-die.

Leclercq, E. y Savonnet, M. (2018). Un modelo de datos basado en tensores para polystore: una aplicación a datos de redes sociales. En *Actas del 22° simposio internacional de aplicaciones e ingeniería de bases de datos (IDEAS 2018), Villa San Giovanni, Italia, págs. 110–118.* https://doi.org/10.1145/3216122.3216152.

Linstedt, D. (2011). Supercargue su almacén de datos: Reglas de modelado de datos invaluables para implementar sus datos. Publicación independiente de Vault CreateSpace.

Maccioni, A. y Torlone, R. (2018). KAYAK: un marco para la preparación de datos justo a tiempo en un lago de datos. En: Conferencia internacional sobre ingeniería de sistemas de información avanzada (CAiSE 2018), Tallin, Estonia, págs. 474–489. https://doi.org/10.1007/978-3-319-91563-0 29.

Madera, C. y Laurent, A. (2016). La próxima evolución de la arquitectura de la información: la ola del lago de datos. En *8ª conferencia internacional sobre gestión de ecosistemas digitales (MEDES 2016), Biarritz, Francia, pp. 174–180.* https://doi.org/10.1145/3012071.3012077.

Maroto, C. (2018). Seguridad del lago de datos: cuatro áreas clave a tener en cuenta al proteger su lago de datos. https://www.searchtechnologies.com/blog/data-lake-security.

Mathis, C. (2017). Lagos de datos. *Datenbank-Spektrum,* 17(3), 289–293. https://doi.org/10.1007/s13222-017-0272-7.

Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., Tekes, S., Riekki, J. (2019). Implementación de big data lake para fuentes de datos heterogéneas. En *2019 IEEE 35th conferencia internacional sobre talleres de ingeniería de datos (ICDEW), págs. 37–44.* https://doi.org/10.1109/ICDEW.2019.00-37.

Miloslavskaya, N. y Tolstoy, A. (2016). Big data, datos rápidos y conceptos de lago de datos. En *la séptima conferencia internacional anual sobre arquitecturas cognitivas inspiradas biológicamente (BICA 2016), Nueva York, EE. UU., Procedia Computer Science, vol. 88, págs. 1 a 6.* https://doi.org/10.1016/j.procs.2016.07.439.

Nargesian, F., Pu, KQ, Zhu, E., Bashardoost, BG, Miller, RJ (2018). Optimización de organizaciones para navegar por lagos de datos. arXiv:1812.07024.

Nogueira, I., Romdhane, M., Darmont, J. (2018). Modelado de metadatos de lago de datos con una bóveda de datos. En *el 22º simposio internacional de aplicaciones e ingeniería de bases de datos (IDEAS 2018), Villa San Giovanni, Italia* (págs. 253-261). Nueva York: ACM.

O'Leary, DE (2014). Incorporación de IA y crowdsourcing en el gran lago de datos. *sistemas inteligentes ieee,* 29(5), 70–73. https://doi.org/10.1109/MIS.2014.82.

Oram, A. (2015). Gestión del lago de datos. Zaloni.

Pathirana, N. (2015). Modelado de datos de patrimonio industrial y cultural. Tesis de maestría, universite lumi ere Lyon 2 Francia.

Quix, C. y Hai, R. (2018). *Lago de datos,* (págs. 1–8). Berlín: Springer International Publishing. https://doi.org/10.1007/978-3-319-63962-8 7-1.

Quix, C., Hai, R., Vatov, I. (2016). Extracción y gestión de metadatos en lagos de datos con GEMMS. *Complex Systems Informatics and Modeling Quarterly,* 9, 289–293. https://doi.org/10.7250/csimq.2016-9.04.

Ravat, F. y Zhao, Y. (2019). Lagos de datos: Tendencias y perspectivas. En *la 30ª conferencia internacional sobre aplicaciones de bases de datos y sistemas expertos (DEXA* (p. 2019). Austria: Linz.

Ravat, F. y Zhao, Y. (2019). Gestión de metadatos para lagos de datos. En *la 23ª conferencia europea sobre avances en bases de datos y sistemas de información (ADBIS* (p. 2019). Eslovenia: Bled.

Russom, P. (2017). Propósitos, prácticas, patrones y plataformas de los lagos de datos. Investigación TDWI.

Sawadogo, PN, Kibata, T., Darmont, J. (2019). Gestión de metadatos para documentos de texto en lagos de datos. En *la 21ª conferencia internacional sobre sistemas de información empresarial (ICEIS 2019), Heraklion, Creta, Grecia, págs. 72–83.* https://doi.org/10.5220/0007706300720083.

Sawadogo, PN, Scholly, E., Favre, C., Ferey, E., Loudcher, S., Darmont, J. (2019). Sistemas de metadatos para lagos de datos: modelos y características. En *BI y aplicaciones de big data - ADBIS 2019 Short Papers and Workshop, Bled, Eslovenia.*

Singh, K., Paneri, K., Pandey, A., Gupta, G., Sharma, G., Agarwal, P., Shroff, G. (2016). Fusión bayesiana visual para navegar en un lago de datos. En *la 19ª conferencia internacional sobre fusión de información (FUSION 2016), Heidelberg, Alemania, IEEE, pp. 987–994.*

Sirosh, J. (2016). El lago de datos inteligente. https://azure.microsoft.com/fr-fr/blog/the-intelligent-data-lake/.

Stefanowski, J., Krawiec, K., Wrembel, R. (2017). Exploración de datos complejos y grandes. *Revista internacional de matemáticas aplicadas e informática,* 27 (4), 669–679. https://doi.org/10.1515/amcs-2017-0046.

Stein, B. y Morrison, A. (2014). El lago de datos empresarial: mejor integración y análisis más profundos. Pronóstico de tecnología de PWC. http://www.smallake.kr/wp-content/uploads/2017/03/20170313 074222.pdf.

Suriarachchi, I. y Plale, B. (2016). Sistemas de análisis cruzado: un caso para la procedencia integrada en lagos de datos. En *la 12ª conferencia internacional IEEE sobre e-ciencia (e-ciencia 2016), Baltimore, MD, EE. UU., págs. 349–354.* https://doi.org/10.1109/eScience.2016.7870919.

Terrizzano, I., Schwarz, P., Roth, M., Colino, JE (2015). Disputa de datos: el desafiante viaje desde la naturaleza hasta el lago. En: 7.ª Conferencia bienal sobre investigación de sistemas de datos innovadores (CIDR 2015), Asilomar, CA, EE. UU., págs. 1–9. http://cidrdb.org/cidr2015/Papers/CIDR15 Paper2.pdf.

Tharrington, M. (2017). La guía dzone sobre big data, ciencia de datos y análisis avanzado. Zona D.

La Fundación de Software Apache (2019). Atlas de Apache: marco de metadatos y gobernanza de datos para Hadoop. https://atlas.apache.org/.

Tiao, S. (2018). Almacenamiento de objetos para big data: ¿Qué es? ¿Y por qué es mejor? https://blogs.oracle.com/bigdata/what-is-object-storage .

Zikopoulos, P., deRoos, D., Bienko, C., Buglio, R., Andrews, M. (2015). Big data más allá de la exageración. McGraw Educación de la colina.