Módulo 2. Arquitecturas modernas (data lakes, lakehouses, hybrid models)

1. Arquitecturas de almacenamiento de datos

1. Arquitecturas de almacenamiento de datos

En el módulo anterior trabajamos con los conceptos de *business intelligence*, entendimos qué es un *data warehouse* y analizamos cómo se estructura el modelado OLAP. Ya vimos cómo estas arquitecturas permiten organizar datos para responder preguntas del negocio de manera eficiente, y cómo el diseño multidimensional contribuye a representar los datos de forma lógica, integrada y orientada al análisis.

Ahora bien, ¿qué pasa cuando una organización empieza a manejar datos que no provienen de sistemas transaccionales? ¿Qué herramientas se usan cuando los datos llegan en formatos muy distintos, como imágenes, archivos en crudo o datos generados en tiempo real? En este tipo de escenarios, el *data warehouse* puede no ser suficiente por sí solo, y aparecen nuevas alternativas que amplían las posibilidades de almacenamiento y análisis.

En esta unidad, vamos a trabajar con dos de esas alternativas: los *data lakes* y las *lakehouses*. Veremos qué los caracteriza, qué diferencias tienen con el *data warehouse* que ya conocemos, y qué ventajas ofrecen en distintos contextos. El objetivo es entender cómo se configuran estas arquitecturas modernas y qué aportan a las estrategias actuales de gestión de datos.

Data lakes: características y ventajas

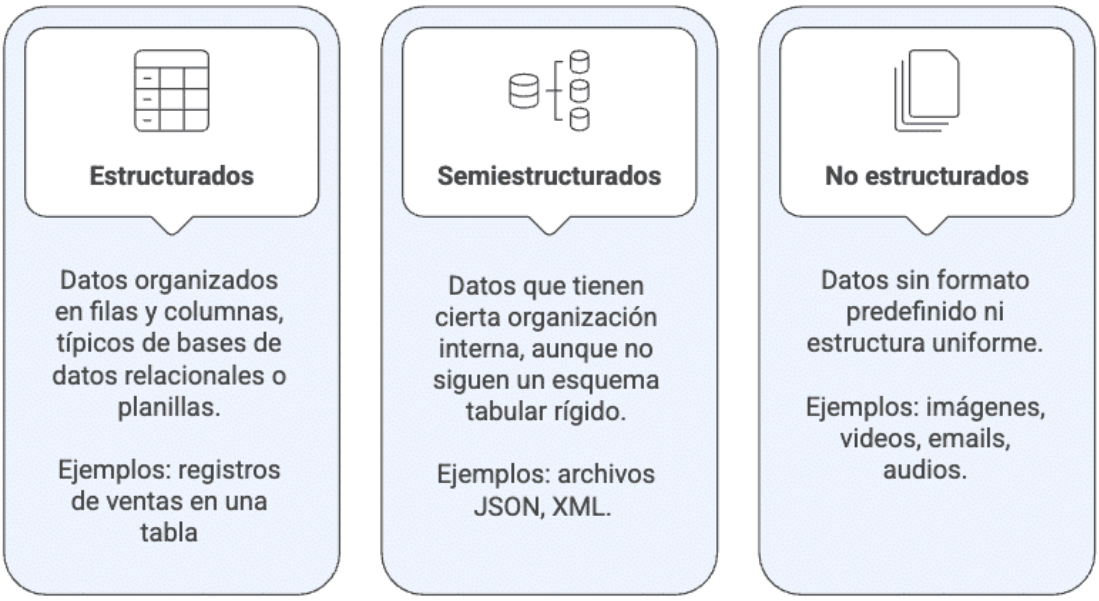
A medida que el volumen, la variedad y la velocidad de los datos aumentaron, las organizaciones comenzaron a enfrentar límites con las soluciones tradicionales de almacenamiento. Durante décadas, las bases de datos relacionales y los *data warehouses* fueron las arquitecturas predominantes. Diseñadas principalmente para datos estructurados, estas soluciones permitían organizar la información de manera consistente y responder preguntas de negocio con precisión. Sin embargo, fueron pensadas en contextos donde el volumen de datos era menor, y los formatos, mucho más homogéneos.

Con el crecimiento exponencial de Internet y el uso masivo de plataformas como redes sociales, servicios de streaming y aplicaciones móviles, las organizaciones empezaron a recibir datos en formatos muy diversos. A diferencia de los registros

tabulares, estos nuevos datos incluían textos libres, imágenes, audios y videos. Este cambio dejó en evidencia las limitaciones de los modelos tradicionales: los esquemas fijos, la necesidad de preprocesamiento y los costos de almacenamiento hacían que no fueran adecuados para absorber este nuevo caudal informativo.

Fue en este contexto que surgió el concepto de *data lake*. James Dixon, director de tecnología en Pentaho, acuñó el término en 2011 para describir un enfoque alternativo al *data warehouse*. La expresión *lake* (lago) remite a la idea de un espacio amplio donde los datos se almacenan en su formato original, sin transformaciones previas. A diferencia del *data warehouse*, que estructura los datos con un modelo específico, el *data lake* permite guardar datos estructurados, semiestructurados y no estructurados en un único repositorio **(Kosinski, 2023).**

Figura 1. Tipos de datos



Fuente: elaboración propia

Ahora bien, ¿en qué más se diferencian estas dos arquitecturas? Si ya trabajamos con el concepto de *data warehouse*, sabemos que está diseñado para ofrecer datos limpios, transformados y organizados bajo una estructura fija. Este tipo de entorno es ideal para generar reportes, tableros de control y análisis de negocio, siempre a partir de información estructurada.

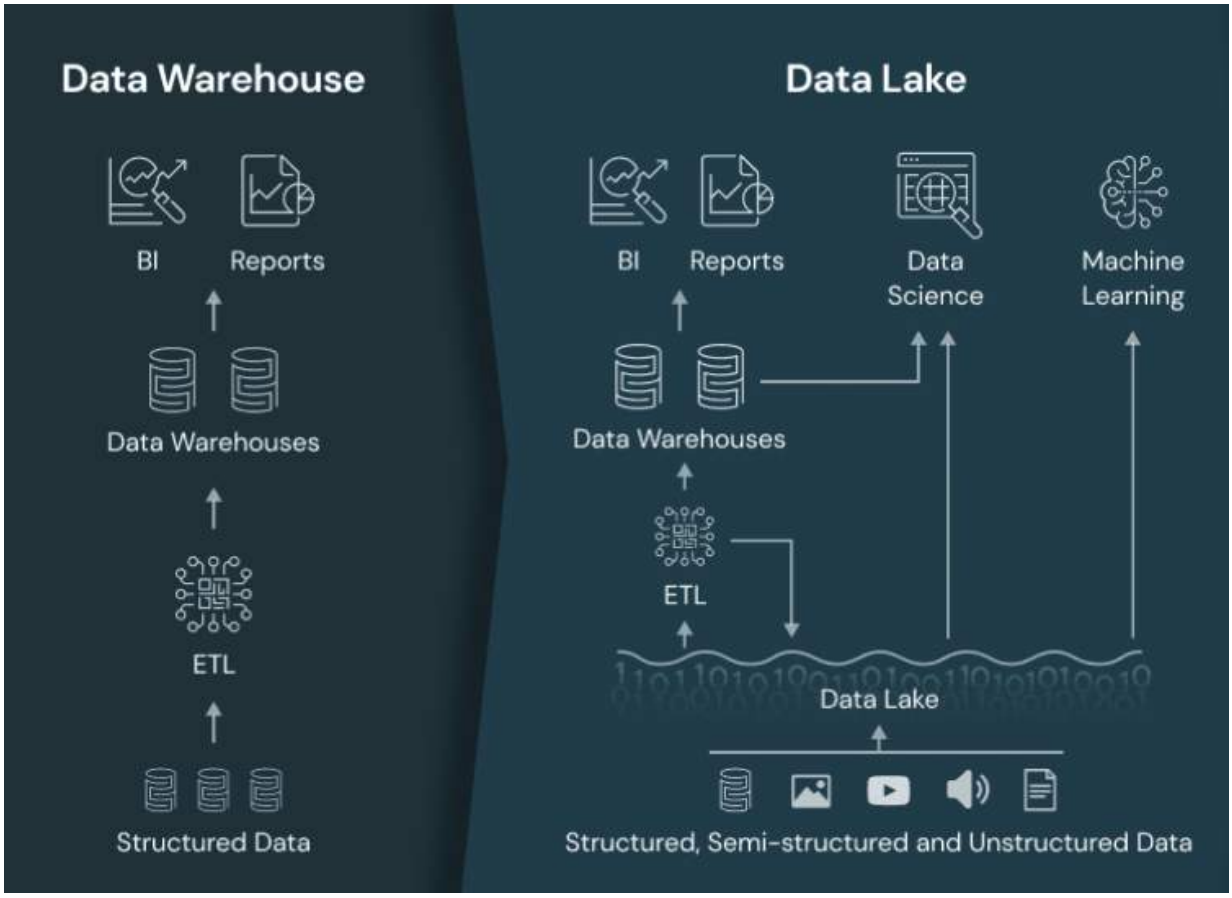
En cambio, el *data lake* fue pensado para almacenar los datos tal como llegan, sin forzar su transformación previa. Esto incluye datos que no siguen un formato tabular, como archivos de texto libre, imágenes, registros de sensores o videos. Esta diferencia no

es solo técnica: tiene un impacto directo en los tipos de uso que permite cada arquitectura.

Mientras que el *data warehouse* se utiliza principalmente para consultas de alto rendimiento y procesos de inteligencia de negocios (BI), el *data lake* permite soportar tareas mucho más diversas, como exploración de datos, ciencia de datos, *machine learning*, entrenamiento de modelos y almacenamiento histórico en gran volumen.

La siguiente figura muestra estas diferencias de forma esquemática:

Figura 2. Diferencia entre *data warehouses* y *data lakes*



Fuente: Fernández, 2025, https://goo.su/8o7JE

Como se observa en la figura, el *data warehouse* sigue un flujo más estructurado: se cargan datos estructurados, se aplican procesos ETL (extracción, transformación y carga) —concepto que desarrollaremos en el siguiente módulo— y luego esos datos son utilizados por herramientas de BI y generación de reportes. En el caso del *data lake*, los datos pueden ingresar sin transformar, en múltiples formatos, y estar disponibles para una variedad de herramientas: desde BI hasta ciencia de datos o algoritmos de aprendizaje automático. Además, el *data lake* puede alimentar a

un *data warehouse* si se requiere, ya que actúa como un repositorio mucho más amplio y flexible.

Los primeros *data lakes* comenzaron a construirse sobre Apache Hadoop, una plataforma de *software* libre que permitía procesar grandes volúmenes de datos distribuidos. En ese momento, se alojaban mayormente en infraestructuras locales, lo que pronto se volvió una limitación a medida que el volumen de datos crecía. La necesidad de escalar rápidamente llevó a muchas organizaciones a migrar sus *data lakes* hacia servicios de almacenamiento en la nube, más flexibles y económicos. Esta transición no solo resolvió el problema del crecimiento, sino que abrió la puerta a nuevas capacidades. Hoy, los *data lakes* no se limitan a almacenar datos de forma masiva: incorporan herramientas para la gestión de metadatos, catálogos de búsqueda, controles de acceso y gobierno del dato. Además, se integran cada vez más en arquitecturas combinadas, como las *lakehouses*, que buscan reunir lo mejor de ambos mundos: el almacenamiento flexible de un *data lake* y las capacidades analíticas optimizadas de un *data warehouse*.

La arquitectura actual de los *data lakes* se basa en servicios de almacenamiento de objetos en la nube, como Amazon S3, Azure Blob Storage o Google Cloud Storage. Estas plataformas permiten almacenar grandes volúmenes de datos sin procesar en múltiples formatos, de forma escalable y con un modelo de costos por

demanda. A diferencia del enfoque tradicional basado en Hadoop, esta arquitectura ofrece mayor flexibilidad operativa y se adapta mejor a las necesidades cambiantes de almacenamiento de datos.

Una característica distintiva de los *data lakes* es la separación entre almacenamiento y procesamiento. Los datos se guardan en bruto en el repositorio central y se procesan solo cuando es necesario, utilizando herramientas externas como Apache Spark. Este modelo técnico habilita el uso del enfoque ELT (*extract, load, transform*) —que también desarrollaremos en el módulo siguiente — en el cual los datos se transforman después de haber sido almacenados, lo que permite trabajar con información cuyo uso aún no está completamente definido. Este principio se conoce como «esquema en lectura» (*schema-on-read*).

Entre sus beneficios más destacados se encuentran la capacidad de integrar datos de múltiples formatos, el desacople entre almacenamiento y procesamiento, los bajos costos de operación y la mejora en el acceso compartido a la información. Este último punto es especialmente relevante para superar los llamados «silos organizacionales»: situaciones en las que cada área

gestiona sus propios datos de forma aislada, sin compartirlos con el resto de la organización. Al centralizar la información en un único entorno accesible, los *data lakes* permiten que distintos equipos trabajen de manera coordinada sobre la misma fuente de datos. En este sentido, se consolidan como una solución estratégica en contextos donde la diversidad y el volumen de datos siguen creciendo.

Lakehouses: combinación de *data lakes* y *data warehouses*

En los últimos años, muchas organizaciones empezaron a enfrentarse a un nuevo tipo de problema: por un lado, necesitaban la flexibilidad de los *data lakes* para almacenar todo tipo de datos; por el otro, requerían el rendimiento y la organización que ofrece un *data warehouse* para realizar análisis rápidos y estructurados. En lugar de elegir entre una u otra arquitectura, comenzó a consolidarse un enfoque que combina lo mejor de ambas: los *data lakehouses*.

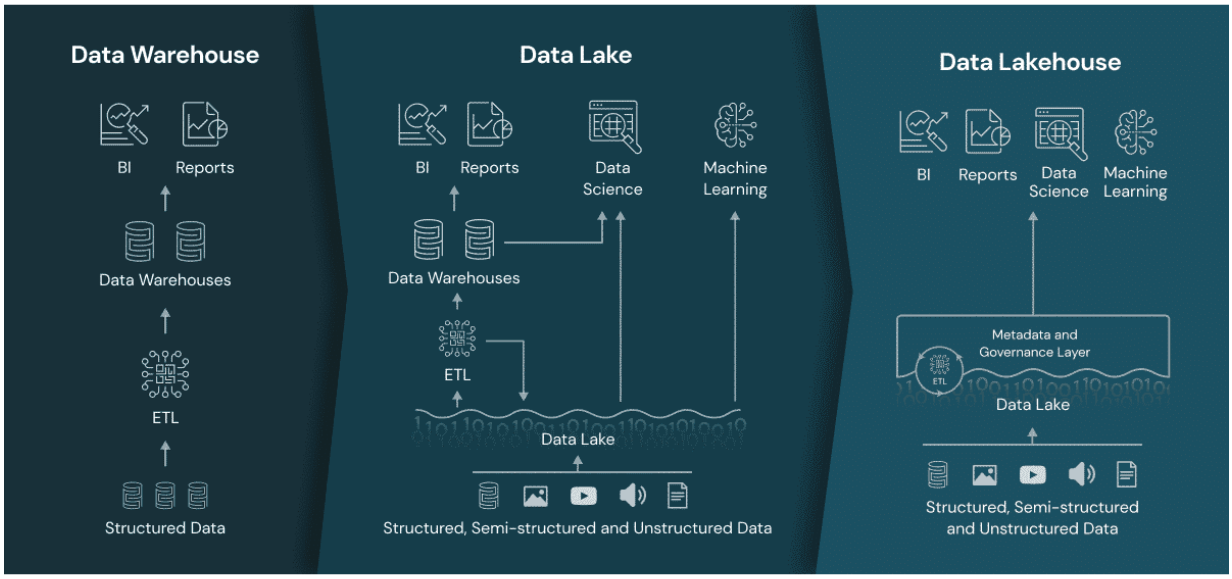
El término *lakehouse* refiere precisamente a esa integración. Se trata de una arquitectura que mantiene la capacidad de almacenar grandes volúmenes de datos en múltiples formatos —como lo hace un *data lake*—, pero que incorpora una capa estructural adicional que permite ejecutar análisis eficientes, con rendimiento comparable al de un *data warehouse*. Esta capa

incluye control de versiones, gestión de metadatos, catálogos de datos y reglas de gobernanza.

El surgimiento de los *lakehouses* responde a la necesidad de unificar los entornos de almacenamiento y análisis. En los modelos anteriores, los datos se cargaban en un *data lake* y, cuando era necesario analizarlos, se movían o duplicaban en un *data warehouse*. Esto implicaba procesos costosos, más infraestructura y mayores tiempos de espera. En cambio, el *lakehouse* permite hacer todo en un mismo entorno, sin necesidad de trasladar los datos.

Retomemos ahora la **Figura 2**, donde ya analizamos las diferencias entre *data warehouses* y *data lakes*. Esta vez, sumamos la tercera arquitectura: el *data lakehouse*. Como se puede ver, el *lakehouse* conserva la base del *data lake* (almacenamiento de datos estructurados, semiestructurados y no estructurados), pero incorpora una **capa de metadatos y gobernanza** que permite organizar los datos para análisis estructurados, sin perder flexibilidad. De este modo, puede responder a múltiples necesidades desde un único repositorio.

Figura 3. Diferencia entre *data warehouses* y *data lakes* y *data lakehouses*



Fuente: Fernández, 2025, https://goo.su/8o7JE

Mientras que el *data warehouse* está optimizado para reportes de negocio y análisis tradicionales, y el *data lake* para almacenamiento masivo y exploración avanzada, el *lakehouse* permite combinar ambos enfoques. Tal como se ve en la figura, esta arquitectura parte de un *data lake* capaz de almacenar datos estructurados, semiestructurados y no estructurados, pero le suma una **capa de metadatos, gobernanza y manejo transaccional** que antes solo existía en los *warehouses*. Gracias a esta integración, en un mismo entorno es posible ejecutar tareas de business intelligence, generar reportes, desarrollar modelos de machine learning y trabajar con ciencia de datos sin mover ni duplicar la información.

Una de las claves del *lakehouse* es que mantiene el enfoque de «esquema en lectura» propio de los *data lakes* —permitiendo cargar datos masivos sin predefinir su estructura—, pero sobre esa base incorpora capacidades transaccionales. Esto habilita operaciones como lecturas consistentes, actualizaciones parciales y control de versiones. Por ejemplo, si en un conjunto de ventas se detecta un error, es posible corregir solo el fragmento afectado sin recargar toda la tabla y manteniendo un historial de cambios para auditorías o comparaciones temporales (Google Cloud, s.f.).

Desde el punto de vista técnico, los *lakehouses* funcionan sobre **motores de consulta analítica** diseñados para procesar datos a gran escala de manera eficiente. Estos motores —como Databricks SQL, Presto, Trino o Dremio— trabajan directamente sobre formatos columnar como Parquet o Avro, optimizados para lectura rápida y compresión. Además, integran seguridad, control de calidad y mecanismos de gobierno del dato, lo que permite utilizar el *lakehouse* como una plataforma única, confiable y de alto rendimiento para toda la analítica de la organización.

En cuanto a los usos concretos, los *lakehouses* son especialmente útiles cuando una organización necesita ejecutar análisis tradicionales y análisis avanzados sobre una misma base de datos. Por ejemplo, pueden alimentar tanto un tablero de visualización para reportes de ventas como un modelo predictivo

de abandono de clientes, sin necesidad de duplicar la información ni mantener dos sistemas paralelos. Esta capacidad de integrar distintos tipos de carga de trabajo en un solo entorno abre la puerta a nuevas configuraciones más flexibles, donde conviven sistemas locales, servicios en la nube y soluciones mixtas. Este enfoque es el que nos lleva al siguiente bloque: **los modelos híbridos y las tendencias actuales en almacenamiento de datos**.

CONTINUAR

2. Modelos híbridos y tendencias en almacenamiento de datos

2. Modelos híbridos y tendencias en almacenamiento de datos

En la unidad anterior trabajamos dos arquitecturas cada vez más comunes en los entornos analíticos actuales: *data lakes* y *lakehouses*. Vimos que permiten almacenar datos en distintos formatos, escalar de manera flexible y adaptarse a diferentes usos, desde tableros de BI hasta modelos de aprendizaje automático. Pero hasta ahora no hablamos de algo central: **¿cómo se insertan estas arquitecturas en los entornos reales de las organizaciones?**

Hoy en día, muchas de estas soluciones se implementan en la nube, porque ofrecen mayor escalabilidad, menor mantenimiento y modelos de pago por uso. Sin embargo, las organizaciones no parten de cero: ya cuentan con infraestructura previa, datos sensibles en servidores propios, o aplicaciones

críticas que siguen funcionando *on-premise*. En este escenario, aparece una necesidad concreta: hacer convivir ambos mundos sin reemplazar uno por otro.

Ahí es donde cobran relevancia los modelos híbridos. No son una herramienta específica, sino una e**strategia para integrar sistemas locales y soluciones en la nube.** Por ejemplo, una empresa puede tener un *data warehouse* tradicional en sus servidores y, al mismo tiempo, montar un *data lake* en la nube para nuevos proyectos analíticos. El valor está en conectarlos: que los datos fluyan, se complementen y puedan aprovecharse sin duplicar esfuerzos.

En esta unidad, primero, vamos a ver cómo funcionan estos modelos híbridos y qué tipo de integraciones son posibles entre sistemas *on-premise* y *cloud*. Después, nos vamos a detener en algunas tendencias actuales en arquitectura de datos, que ayudan a pensar hacia dónde se están moviendo las decisiones tecnológicas en este campo.

Modelos híbridos: integración de sistemas *on-premise* y *cloud*

A la hora de diseñar una arquitectura de almacenamiento de datos, las organizaciones enfrentan una decisión: **¿dónde alojar los datos y los sistemas que los procesan?** En términos

generales, hay dos grandes opciones: mantener todo en infraestructuras internas (*on-premise*) o migrar a servicios externos en la nube (*cloud*). Cada enfoque tiene ventajas y desafíos, y entender sus diferencias es fundamental para pensar una arquitectura híbrida.

Cuando hablamos de entornos *on-premise*, nos referimos a sistemas que están **físicamente alojados en los servidores propios de la organización**. Esto implica que la empresa es responsable de toda la infraestructura: desde el *hardware* y el almacenamiento, hasta el mantenimiento, la seguridad y las actualizaciones. La principal ventaja es el control total sobre los recursos y la ubicación de los datos, lo cual puede ser clave en sectores regulados o con requerimientos estrictos de confidencialidad. Sin embargo, también implica **mayores costos fijos, tiempos más largos de implementación y menor flexibilidad para escalar**.

En cambio, los entornos *cloud* funcionan sobre plataformas ofrecidas por terceros como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform. La infraestructura física está gestionada por el proveedor, y las organizaciones acceden a ella a través de Internet, pagando solo por los recursos que usan. Este modelo permite **escalar rápida y fácilmente**, reducir los costos iniciales y acceder a tecnologías avanzadas sin necesidad de grandes inversiones. A cambio, se depende de un proveedor

externo y es necesario asegurar que los datos estén protegidos según las políticas de la empresa.

Para visualizar mejor estas diferencias, observemos la siguiente tabla:

Tabla 1. Diferencias entre entornos *on-premise* y *cloud base*

Característica	*On-premise*	*Cloud*
Infraestructura	Servidores propios	Infraestructura del proveedor
Costos iniciales	Altos (compra y mantenimiento)	Bajos (modelo de pago por uso)
Escalabilidad	Limitada	Alta y flexible
Tiempo de implementación	Lento	Rápido
Mantenimiento	Interno (requiere personal dedicado)	Externo (lo gestiona el proveedor)
Control sobre los datos	Total (localizados en la organización)	Parcial (dependencia del proveedor)

Actualización tecnológica	Requiere inversión propia	Acceso inmediato a tecnologías avanzadas

Fuente: Genese Solution, 2022, https://goo.su/UnR5Z

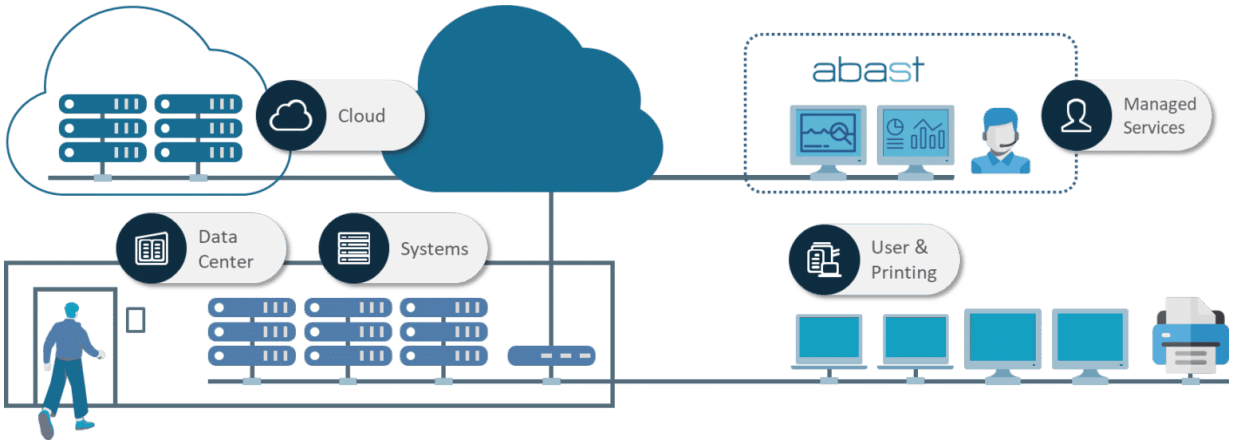
En muchos casos, optar exclusivamente por una solución *on-premise* o completamente en la nube no es lo más conveniente. Las organizaciones suelen tener infraestructura existente, sistemas heredados que siguen operativos o requerimientos normativos que obligan a mantener ciertos datos en servidores propios. A la vez, las soluciones en la nube permiten escalar con rapidez, reducir costos de implementación y acceder a tecnologías avanzadas sin grandes inversiones.

Por eso, se vuelve habitual adoptar un enfoque combinado. **El modelo híbrido permite conservar lo que ya funciona en el entorno local**, y sumar capacidades nuevas a través de servicios en la nube, como almacenamiento adicional, procesamiento intensivo o herramientas de análisis avanzado. Este enfoque brinda flexibilidad y se adapta mejor a las necesidades concretas de cada organización.

Veamos la siguiente figura que representa el funcionamiento de una arquitectura híbrida. Este enfoque combina recursos locales

(*on-premise*), como centros de datos propios, con servicios y almacenamiento en la nube.

Figura 4. Arquitectura híbrida



Fuente: Abast, 2024, https://goo.su/sAUhjH

En este esquema, lo que vemos es una **infraestructura híbrida** que conecta un centro de datos local con servicios en la nube. Cada entorno cumple un rol distinto: los sistemas ***on-premise***, es decir, aquellos que están instalados físicamente en la organización, suelen usarse para alojar información sensible o para ejecutar procesos que necesitan **respuesta inmediata**. A esto se lo llama **baja latencia**, y significa que entre que se realiza una acción y se obtiene una respuesta pasa muy poco tiempo. Por su parte, la **nube** se utiliza para tareas que requieren mayor

capacidad de almacenamiento, procesamiento flexible o acceso a tecnologías que no están disponibles localmente.

Ambos entornos están conectados y pueden **intercambiar datos o tareas** de forma coordinada. Por ejemplo, una empresa puede gestionar su sistema de facturación desde servidores propios, pero enviar automáticamente los datos de ventas a la nube para generar reportes, analizar tendencias o entrenar un modelo de predicción de demanda. Así, se aprovechan las ventajas de los dos entornos sin tener que elegir uno solo.

Una ventaja del enfoque híbrido es que permite distribuir los recursos tecnológicos según lo que se necesita en cada caso. Por ejemplo, una empresa del sector financiero puede mantener sus sistemas transaccionales en servidores propios, especialmente si tiene requisitos normativos estrictos, y al mismo tiempo aprovechar la nube para almacenar datos históricos o ejecutar modelos de *machine learning*. Esta combinación ayuda a aprovechar lo que ya está instalado y, a la vez, incorporar capacidades nuevas.

También es útil cuando hay cargas de trabajo variables. Muchas organizaciones no tienen una demanda constante de procesamiento: algunos períodos requieren más potencia, como el cierre de un trimestre o una campaña comercial. En esos casos, se puede usar la nube solo cuando hace falta más capacidad, sin

tener que invertir en más servidores. Esta práctica se conoce como *cloud bursting* y permite ampliar el procesamiento solo por el tiempo necesario.

En este marco, donde la flexibilidad y la escalabilidad se vuelven fundamentales, empiezan a consolidarse nuevas formas de pensar las arquitecturas de datos. Modelos distribuidos, soluciones especializadas y enfoques más ágiles aparecen como respuesta a las demandas actuales. A continuación, veremos algunas de las tendencias emergentes que están transformando la manera en que las organizaciones gestionan y utilizan sus datos.

Tendencias emergentes en arquitecturas de datos

A medida que crecen las demandas sobre los datos —ya sea en volumen, velocidad o diversidad de usos—, también evolucionan las formas de organizarlos y gestionarlos. En los últimos años, surgieron nuevas propuestas que buscan superar los límites de los modelos tradicionales, especialmente aquellos centralizados y difíciles de escalar. Estas tendencias emergentes responden a una necesidad común: acercar los datos a quienes los usan,

facilitar el acceso en tiempo real y adaptarse a entornos cada vez más distribuidos, dinámicos y complejos.

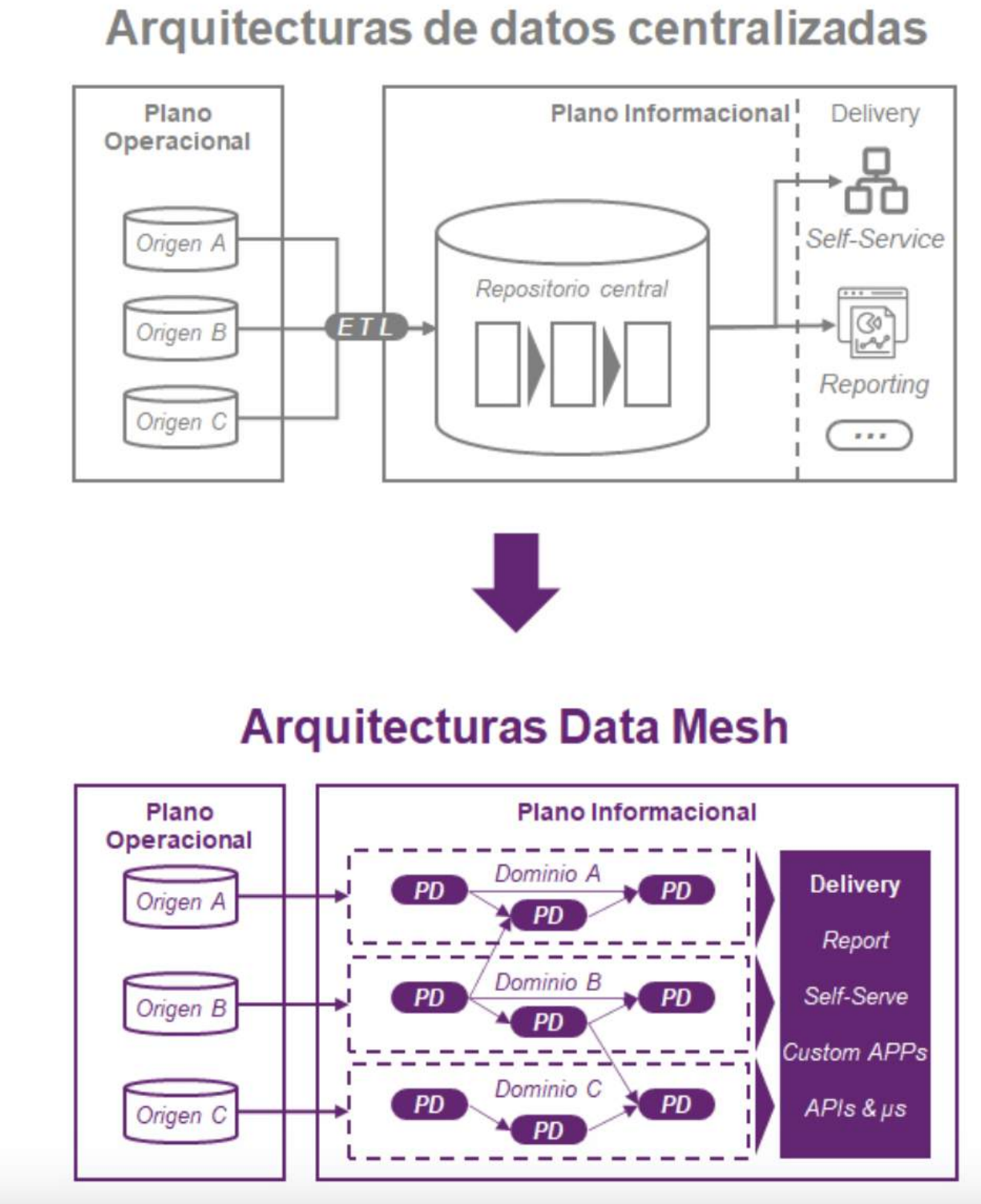
En este contexto, aparecen conceptos como *data mesh* y *edge computing*, que apuntan a una gestión más ágil, descentralizada y contextual de los datos. Mientras que el *data mesh* promueve la responsabilidad distribuida dentro de la organización, el *edge computing* lleva el procesamiento directamente al lugar donde se generan los datos, reduciendo los tiempos de respuesta. A continuación, nos centraremos en explicar estas dos propuestas y en qué tipo de situaciones pueden ser útiles.

Data mesh: hacia una arquitectura descentralizada

Una situación frecuente en muchas organizaciones es la dificultad para acceder a los datos cuando se los necesita. Aunque existan repositorios bien organizados, cada vez que un equipo quiere obtener información específica suele depender del área de IT, generando demoras, pedidos cruzados y cuellos de botella. Este problema no siempre tiene que ver con la falta de datos, sino con cómo están organizados y quién puede acceder a ellos. En este contexto, surge el enfoque ***data mesh***, una forma diferente de gestionar los datos que busca hacer más eficiente su uso diario dentro de las organizaciones.

Para entender en qué consiste, podemos observar la siguiente figura.

Figura 5. Comparación entre arquitectura centralizada y arquitectura *data mesh*



Fuente: Grande y Labella, 2022, https://goo.su/pFLQfeL

En la parte superior de la figura se representa el **modelo tradicional de arquitectura de datos centralizada**. En este enfoque, los datos generados por distintas fuentes —representadas como A, B y C— son recolectados y enviados a un único repositorio central mediante procesos de integración conocidos como ETL (como dijimos anteriormente, desarrollaremos este concepto en el próximo módulo). Este repositorio actúa como una única fuente de verdad, donde todo el procesamiento, almacenamiento y control de calidad es gestionado por un equipo técnico centralizado. Desde allí, los usuarios acceden a los datos a través de herramientas de análisis o servicios de autoservicio preconfigurados. Aunque este modelo asegura consistencia, también puede generar cuellos de botella, ya que todo pasa por un mismo equipo, lo que puede ralentizar los tiempos de respuesta.

En la parte inferior, se ilustra el enfoque propuesto por *data mesh*. A diferencia del modelo anterior, acá los datos no se agrupan en un único repositorio, sino que se **organizan por dominios**: es decir, cada área de la organización (por ejemplo, ventas, finanzas o recursos humanos) se convierte en **responsable de sus propios datos**. Cada dominio se encarga de recolectar, procesar y ofrecer sus datos como un producto listo

para ser utilizado por otros. Esto implica que los datos se mantienen cerca del lugar donde se generan y se gestionan con conocimiento del negocio, lo que mejora su relevancia y accesibilidad. Además, la gobernanza —es decir, las reglas sobre seguridad, calidad y acceso— también se distribuye entre los dominios, permitiendo mayor autonomía sin perder el control general.

Para entender este proceso en la práctica, pensemos en el caso de una empresa de comercio electrónico. En un enfoque *data mesh*, el **equipo de ventas** podría ser responsable de los datos relacionados con transacciones, carritos abandonados y promociones. Por su parte, el **equipo de logística** gestionaría los datos de envíos, tiempos de entrega y devoluciones, mientras que **recursos humanos** manejaría la información sobre empleados, contrataciones y ausentismo. Cada uno de estos equipos (o dominios) administra sus propios datos como un producto: se encargan de capturarlos, asegurarse de su calidad y compartirlos con el resto de la organización mediante interfaces estándar.

Por ejemplo, si el área de finanzas necesita calcular el costo real de una operación, puede acceder directamente a los datos del dominio de logística (para conocer el gasto de envío) y del dominio de ventas (para conocer el monto cobrado), sin necesidad de que un equipo técnico centralizado tenga que

extraer, combinar y entregar esa información. Esto agiliza la toma de decisiones, reduce la carga sobre los equipos técnicos y mejora la fluidez del trabajo entre áreas.

Este cambio de paradigma tiene varios objetivos. Por un lado, permite que los equipos trabajen de forma más independiente, sin tener que esperar a que un área técnica les prepare los datos. Por otro, mejora la colaboración entre sectores, ya que la información se publica como un recurso compartido, estandarizado y fácil de consumir. Así, un equipo de *marketing* puede utilizar datos de ventas o logística sin depender de múltiples intermediarios o sin tener que solicitar informes a medida cada vez que lo necesite (Data Quality, 2025).

Uno de los conceptos centrales del *data mesh* es tratar los datos como productos. Esto significa que cada conjunto de datos debe estar bien definido, documentado y disponible como si fuera un servicio. Cada dominio —recordemos que por dominio entendemos a cada área de la organización– se convierte en productor y consumidor de datos, generando un ecosistema donde la información circula con fluidez, pero bajo ciertos

estándares de calidad, accesibilidad y seguridad definidos por quienes los producen.

Otro principio importante es el **enfoque de gobierno del dato distribuido o *bottom-up*.** En lugar de que todas las políticas de acceso, calidad o seguridad se definan desde un único equipo central, son los propios dominios quienes establecen las reglas sobre cómo se producen, consumen y protegen sus datos. Esto no significa perder el control general, sino permitir que las decisiones estén más cerca de quienes conocen mejor el uso que se le da a la información. Por ejemplo, en una empresa de salud, el equipo de recursos humanos puede definir sus propias reglas para el manejo de datos de empleados, como quién puede ver información sobre licencias médicas o evaluaciones de desempeño. Mientras tanto, el área de finanzas establece otras políticas para proteger datos sensibles como salarios o presupuestos. Cada dominio aplica medidas de seguridad y calidad acordes a su contexto, sin necesidad de que todas las decisiones pasen por un comité central. Esto agiliza el trabajo, reduce burocracia y asegura que quienes están más cerca de los datos tomen decisiones informadas sobre su uso.

Por último, es importante aclarar que el *data mesh* no reemplaza por completo a los modelos anteriores, como los *data warehouses* o data lakes. En muchos casos, estos siguen cumpliendo funciones importantes, como el almacenamiento centralizado o

el respaldo histórico. Lo que propone el *data mesh* es sumar una capa organizativa que permita a las empresas trabajar con datos de forma más ágil, flexible y cercana a las necesidades reales de cada equipo. En este sentido, representa una evolución natural para organizaciones que quieren aprovechar mejor sus recursos informativos sin sobrecargar sus estructuras técnicas.

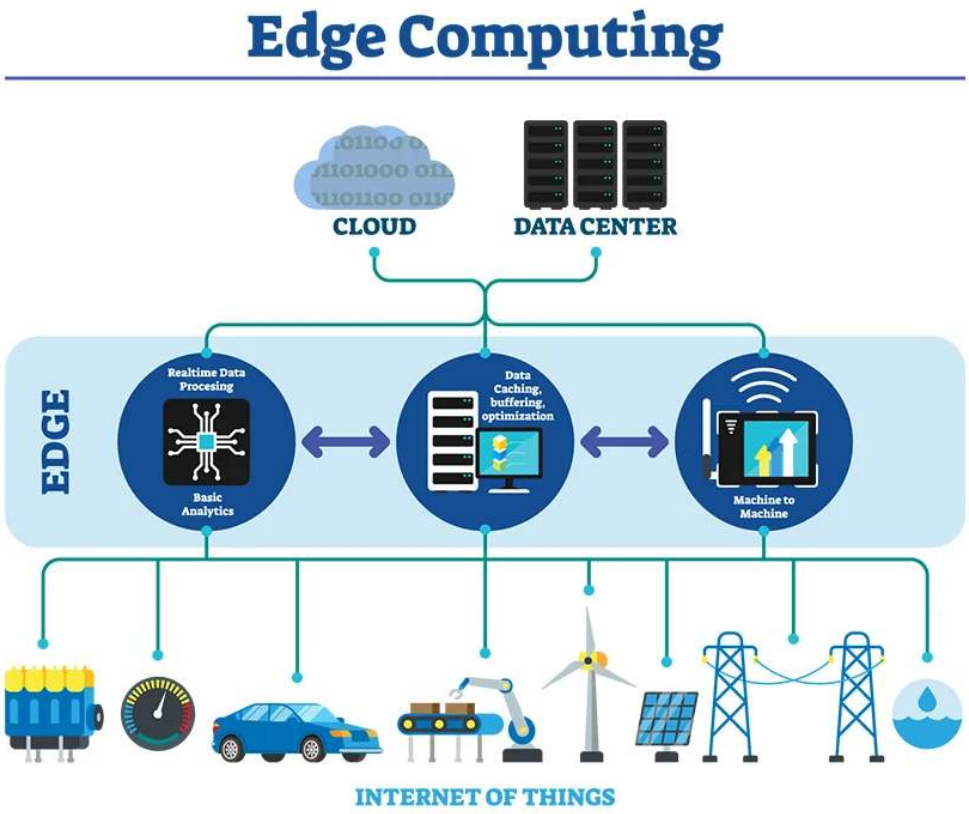
Edge computing: procesamiento cerca del origen de los datos

En el contexto de arquitecturas de datos modernas, una de las tendencias emergentes más relevantes es el *edge computing*. Este enfoque busca resolver un problema que aparece con el crecimiento del *Internet of Things* (IoT) y el volumen masivo de datos que se generan fuera del centro de datos tradicional: ¿cómo procesar información de manera más ágil, sin saturar redes ni depender por completo de la nube?

A diferencia del modelo clásico en el que los datos viajan desde los dispositivos hacia un centro remoto (la nube o un *data center*) para su procesamiento, *edge computing* propone realizar parte de ese procesamiento en el borde de la red, es decir, lo más cerca posible del lugar donde se generan los datos. Esto permite tomar decisiones más rápidas, reducir los tiempos de respuesta (latencia) y optimizar el uso de los recursos.

La siguiente figura ilustra este funcionamiento de forma clara:

Figura 6. Funcionamiento del enfoque de *edge computing* en entornos de Internet de las Cosas



Fuente: Taikun, s.f., https://goo.su/oJUrtc

En la parte inferior, se muestra una variedad de dispositivos conectados a internet —desde autos y turbinas eólicas, hasta sensores industriales—, todos ellos parte del ecosistema IoT. Esos dispositivos están conectados a una capa intermedia llamada *«edge»* , donde se realizan tareas como análisis básico y procesamiento en tiempo real. Esta es una de las claves del enfoque: no todos los datos necesitan viajar hasta la nube.

Dentro de esa capa intermedia (resaltada en celeste), se representan tres etapas: primero, el procesamiento en tiempo real y análisis básico; luego, el almacenamiento en caché y la optimización de datos; y finalmente, la comunicación entre máquinas (*machine-to-machine*), que permite que los dispositivos interactúen entre sí sin intervención humana. Recién después de esas etapas, si es necesario, los datos son enviados a la nube o al centro de datos para tareas más complejas, almacenamiento a largo plazo o análisis agregados.

Este tipo de arquitectura es especialmente útil en contextos donde cada milisegundo cuenta. Por ejemplo, en una fábrica automatizada, un sensor que detecta una falla no puede esperar a que la información viaje a la nube, se procese y vuelva: necesita reaccionar en el momento. Lo mismo ocurre con un auto conectado que recibe datos del entorno para frenar ante un obstáculo. Procesar esa información en el borde permite evitar demoras que podrían afectar el funcionamiento o la seguridad del sistema.

Desde el punto de vista de *business intelligence*, el *edge computing* abre la puerta a nuevas posibilidades: habilita el análisis distribuido —es decir, permite que los datos se procesen cerca del lugar donde se generan, en lugar de enviarlos todos a un servidor central—, mejora la eficiencia operativa y permite tomar decisiones basadas en datos en tiempo real. No reemplaza a la

nube, sino que la complementa, aliviando la carga y priorizando lo que realmente debe enviarse a instancias centrales. Además, mejora la privacidad, ya que los datos sensibles pueden mantenerse localmente sin necesidad de ser transmitidos.

Estas son solo algunas de las tendencias que empiezan a marcar el rumbo en el diseño de arquitecturas de datos. A medida que las organizaciones generan y consumen cada vez más información, surgen nuevas necesidades que impulsan el desarrollo de soluciones innovadoras. Por eso, es esperable que continuamente aparezcan nuevos enfoques y herramientas que busquen responder a los desafíos que plantea este crecimiento. Adaptarse a estas transformaciones será clave para aprovechar el valor de los datos en contextos cada vez más dinámicos y exigentes.

CONTINUAR

Referencias

Abast, (2024). *Hybrid Cloud: La mejor solución entre un entorno on-premise y la nube pública.* https://www.abast.es/blog/hybrid-cloud-la-mejor-solucion-entre-un-entorno-on-premise-y-la-nube-publica/

Data Quality, (2025). *Data Mesh: qué es, ventajas y cómo implementar esta arquitectura de datos.* https://www.elternativa.com/data-mesh/

Google Cloud, (s.f.). *¿Qué es data lakehouse?* https://cloud.google.com/discover/what-is-a-data-lakehouse?hl=es-419

Grande, C., & Labella, E. (2022). *¿Qué es un Data Mesh y cómo saber si tu empresa lo necesita?* KPMG Tendencias. https://www.tendencias.kpmg.es/2022/04/data-mesh-descentralizacion-organizativa-informacion/

Fernández, O. (2025). *Data Lakehouse: La solución híbrida para Big Data.* https://aprenderbigdata.com/lakehouse/

Kosinski, M. (2023). ¿Qué es un data lake? IBM. https://www.ibm.com/es-es/think/topics/data-lake

Taikun, (s.f.). *How Does Edge Computing Technology Work in Simple Terms.* https://taikun.cloud/how-does-edge-computing-technology-work-in-simple-terms/

CONTINUAR

Descarga en PDF