Módulo 3. Procesos ETL y ELT, integración cloud (Snowflake, BigQuery, Databricks)

- Procesos de integración de datos: ETL y ELT
- Plataformas cloud para integración de datos
- Referencias

Procesos de integración de datos: ETL y ELT

En los módulos anteriores ya recorrimos los principales entornos donde se almacenan y gestionan los datos: el *data warehouse* como estructura ordenada y optimizada para el análisis; el *data lake*, con su enfoque más flexible y bruto; y el *data lakehouse*, como propuesta híbrida. También nos detuvimos en los conceptos centrales de *business intelligence*, entendiendo cómo la información organizada y accesible puede contribuir a una mejor toma de decisiones.

Ahora bien, una pregunta que suele quedar flotando es cómo llegan los datos hasta esos entornos, y cómo se transforman desde su forma original —muchas veces desestructurada y dispersa— hasta convertirse en información útil para analizar. Estos interrogantes nos llevan directamente a los procesos de integración de datos, entre los cuales dos tienen un rol central: ETL y ELT.

En esta unidad vamos a trabajar ambos procesos por separado. Primero abordaremos el proceso ETL —extracción,

transformación y carga—, que ha sido el enfoque tradicional durante décadas. Luego nos centraremos en ELT —extracción, carga y transformación—, un enfoque que ha ganado protagonismo con la expansión de las plataformas en la nube. Más allá de las siglas, lo importante es comprender en qué se diferencian, qué decisiones técnicas y de arquitectura implican, y en qué contextos conviene utilizar uno u otro. El objetivo es que puedas reconocerlos, evaluarlos y, llegado el caso, aplicarlos como parte de una estrategia de integración de datos sólida y alineada con las necesidades de un entorno de *business intelligence*.

ETL: extracción, transformación y carga tradicional

Antes de que los datos se conviertan en informes, visualizaciones o tableros interactivos, hay una etapa indispensable que garantiza que esa información tenga sentido y sea confiable. Nos referimos al proceso que permite preparar los datos para que sean utilizables en entornos de análisis. Ese proceso es conocido como ETL —sigla en inglés de *extract, transform and load*— y consiste en una secuencia ordenada de pasos técnicos y lógicos. A través de este procedimiento, los datos se extraen desde múltiples fuentes, se limpian, se organizan y se cargan en un

sistema donde pueden ser consultados, relacionados y utilizados para distintos fines analíticos.

Figura 2. Proceso ETL

Fuente: elaboración propia con base en IBM, 2024

La primera fase del proceso ETL es la **extracción**. Esta etapa consiste en recuperar los datos desde sus fuentes de origen y llevarlos hacia una zona de preparación, sin modificar los sistemas originales. Para ello, es necesario conocer cuáles son esas fuentes y qué características tienen. En general, se puede trabajar con datos provenientes de sistemas estructurados y no estructurados, tanto antiguos como actuales. En general, se puede trabajar con datos provenientes de sistemas estructurados y no estructurados, tanto antiguos como actuales.

Las herramientas de extracción se conectan a estos sistemas para copiar la información de forma segura y controlada. Estos datos requieren técnicas específicas para ser interpretados correctamente. Las herramientas de extracción se conectan a estos sistemas para copiar la información de forma segura y controlada.

A continuación, se presenta una tabla con algunos de los tipos de fuentes más utilizados en los procesos de extracción, junto con una breve descripción de cada una:

Tabla 1. Tipos de fuentes de datos utilizadas en la extracción

Fuente de datos	Descripción	Tipo de dato
Servidores SQL	Bases de datos estructuradas que organizan la información en tablas relacionales.	Estructurado
Servidores NoSQL	Bases de datos no relacionales, útiles para trabajar con	Semiestructurado / No estructurado

	grandes volúmenes de datos no estructurados o semiestructurados.	
Archivos planos (JSON, XML)	Documentos que almacenan datos en texto plano, estructurados mediante etiquetas o formatos definidos.	Semiestructurado
Sistemas CRM	Plataformas que gestionan la relación con clientes, registrando interacciones, ventas, soporte, etc.	Estructurado
Sistemas ERP	Sistemas de planificación de recursos empresariales que integran áreas como	Estructurado

	finanzas, logística y RR. HH.	
Correos electrónicos	Fuente de datos que puede incluir contenido útil para análisis de comunicación o seguimiento.	No estructurado
Páginas web	Sitios desde los que se extraen datos mediante técnicas como el *web scraping*.	No estructurado

Fuente: elaboración propia

Una vez identificadas y conectadas las fuentes, los datos extraídos se almacenan temporalmente en una zona de preparación o *staging area*. Allí permanecerán aislados del entorno operativo, listos para ser transformados. Esta separación permite trabajar sobre la información sin afectar los sistemas fuente, asegurando la integridad de las operaciones habituales de la organización. Además, en esta etapa se evalúa si los datos extraídos son completos, actuales y adecuados para los objetivos del análisis posterior.

La segunda fase del proceso ETL es la **transformación**. En esta etapa, los datos extraídos —que, como vimos, pueden estar en formatos muy diversos— son procesados para ser útiles y coherentes con los objetivos analíticos. El propósito principal es convertir datos sin procesar en información organizada, depurada y con estructura consistente. Esto se logra a través de una serie de procesos que aplican reglas de negocio, validaciones y modificaciones sobre los datos para adaptarlos al sistema de destino. En la siguiente figura presentamos algunos de estos procesos:

Figura 2. Procesos principales en la fase de transformación

Fuente: elaboración propia

Estos procesos permiten que los datos pasen de su forma original —que puede ser desordenada o incompleta— a un formato estructurado, coherente y listo para ser analizado. Por ejemplo, una empresa que recibe datos de ventas desde distintos puntos de venta puede encontrarse con archivos que tienen distintas monedas, formatos de fecha o incluso nombres de productos escritos de forma inconsistente. Mediante el proceso de transformación, esos datos pueden unificarse: se convierte toda la información a una misma moneda, se normalizan los nombres de productos y se aplican reglas para que todas las fechas tengan el mismo formato. Así, lo que antes era un conjunto de archivos dispersos se convierte en un informe consolidado, útil para evaluar el rendimiento comercial.

La tercera y última fase del proceso ETL es la carga. Una vez que los datos han sido transformados y organizados, deben ser trasladados desde la zona de preparación hacia un sistema donde puedan ser almacenados de forma definitiva y utilizados para distintos fines analíticos. Esta etapa consiste en mover los datos hacia una plataforma de destino previamente definida, como un *data warehouse* o un *data lake*. El objetivo es que queden disponibles para ser consultados por los usuarios, integrados en informes o utilizados en procesos automatizados.

La forma en que se realiza la carga depende de varios factores, como el volumen de datos, la frecuencia de actualización, la infraestructura disponible y las necesidades del análisis. En general, se distingue entre tres tipos de cargas:

Figura 3. Tipos de carga

Fuente: elaboración propia con base en IBM, 2024

Estas operaciones pueden ejecutarse en horarios programados, usualmente fuera de las horas pico, para evitar interferencias con los sistemas operativos.

En esta fase final del proceso ETL es necesario considerar con precisión el sistema de destino donde se alojarán los datos. Como ya se abordó en módulos anteriores, existen diferentes enfoques para el almacenamiento de grandes volúmenes de

información: el *data warehouse*, el *data lake* y el *data lakehouse*. La elección entre uno u otro dependerá del tipo de datos que se estén procesando, de los objetivos analíticos y del diseño general de la arquitectura.

El *data warehouse* continúa siendo la opción más extendida para entornos donde se prioriza el análisis estructurado, con información organizada en esquemas definidos y orientada a consultas analíticas rápidas. Por su parte, el *data lake* ofrece una mayor flexibilidad, permitiendo almacenar datos en su formato original —estructurados, semiestructurados o no estructurados— y procesarlos más adelante según sea necesario. El modelo *data lakehouse*, en cambio, busca combinar lo mejor de ambos mundos: estructura y rendimiento analítico del *data warehouse*, junto con la flexibilidad y escalabilidad del *data lake*.

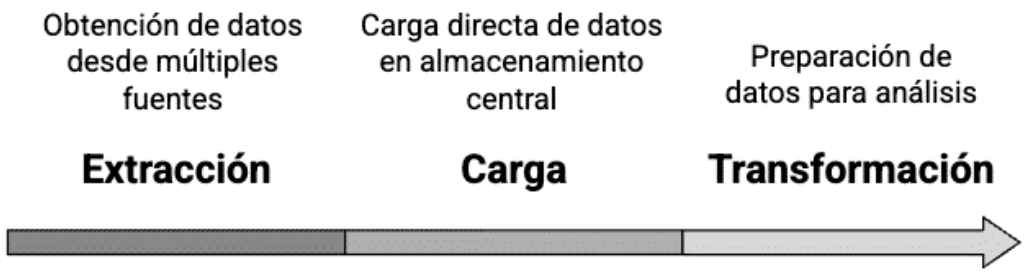
En muchas organizaciones, estos sistemas conviven y se complementan. Los datos pueden ser inicialmente alojados en un *data lake* por su capacidad de absorber múltiples formatos sin necesidad de transformación inmediata, y luego ser integrados en un *data warehouse* o en una arquitectura *lakehouse* para su análisis posterior. Sea cual sea el modelo elegido, lo importante es que la etapa de carga sea capaz de trasladar la información procesada desde la zona de preparación hacia un entorno donde pueda ser consultada, integrada en informes o utilizada en procesos automatizados.

En contextos actuales, esta etapa de carga tiende a estar altamente automatizada. Las herramientas modernas permiten definir flujos de trabajo programados que actualizan la información sin intervención manual, asegurando que los datos estén disponibles, actualizados y alineados con los requerimientos analíticos del negocio. Una carga eficiente es el punto de conexión entre el procesamiento previo y la toma de decisiones basada en datos confiables.

ELT: extracción, carga y transformación en entornos modernos

Con el crecimiento de las plataformas *cloud* y los sistemas distribuidos, el procesamiento de datos adoptó nuevas estrategias que se alejan del modelo tradicional ETL. Uno de los enfoques que ha ganado relevancia en los últimos años es ELT, sigla en inglés de *extract, load and transform*, es decir, extraer, cargar y transformar. Este modelo reorganiza el orden de las operaciones, trasladando la transformación al final del proceso y aprovechando la capacidad de cómputo que ofrecen los entornos de almacenamiento modernos. De esta forma, se plantea una lógica distinta para la integración de datos, más adaptada a infraestructuras en la nube y a cargas de trabajo dinámicas.

Figura 4. Proceso ELT



Fuente: elaboración propia con base en AWS, s.f.

En el modelo ELT, la **extracción** sigue siendo el primer paso. Como ya se explicó, los datos pueden provenir de fuentes estructuradas y no estructuradas, internas o externas. La diferencia central con el modelo anterior es que, en lugar de transformarlos antes de ingresarlos al sistema, en ELT se **cargan** directamente en el repositorio central sin modificaciones previas.

Este repositorio suele estar alojado en una infraestructura cloud, como Snowflake, BigQuery o Databricks —plataformas *cloud* de integración de datos que desarrollaremos en la siguiente unidad —, lo que permite absorber grandes volúmenes de datos en bruto sin generar cuellos de botella. En este sentido, el almacenamiento flexible y la posibilidad de escalar horizontalmente resultan fundamental: los datos llegan al sistema de destino tal como fueron capturados, y allí se

resguardan de forma segura hasta que se los necesite para análisis o procesos posteriores.

Una vez realizada la carga, comienza la etapa de transformación. En este modelo, esa transformación ocurre *dentro* del sistema de almacenamiento, es decir, no se necesita un servidor o motor externo para limpiar, enriquecer o reestructurar los datos. Las plataformas *cloud* modernas están preparadas para ejecutar consultas complejas, aplicar reglas de negocio, unificar formatos y realizar operaciones sobre conjuntos de datos de gran tamaño. Esto no solo simplifica la arquitectura, sino que también reduce los tiempos de transferencia y los costos asociados a la duplicación de procesos.

Una de las principales ventajas del modelo ELT es su capacidad para adaptarse a entornos donde los datos crecen rápidamente, se actualizan con frecuencia y son utilizados por distintos equipos para fines diversos. En lugar de aplicar transformaciones previas, este enfoque permite cargar los datos en su forma original y procesarlos según las necesidades específicas de cada área. Por ejemplo, en una empresa de logística, el área de operaciones puede cargar datos de seguimiento de envíos desde dispositivos IoT directamente en la nube, y luego aplicar transformaciones para generar reportes de cumplimiento, mientras que el área de marketing accede a los mismos datos crudos para analizar patrones de comportamiento de los clientes.

Este tipo de flexibilidad requiere una gestión ordenada. Como los datos no se limpian antes de ser cargados, es necesario definir con claridad qué transformaciones se van a ejecutar, quién las diseña y quién las aprueba. En una organización, esto suele implicar la participación del equipo de ingeniería de datos, que se encarga de diseñar los flujos y escribir los scripts de transformación; del equipo de analistas, que define qué reglas de negocio se deben aplicar; y del área de gobernanza de datos, que valida la calidad y consistencia de los resultados. Si estos roles no están bien definidos, pueden surgir errores como métricas duplicadas, reportes inconsistentes o dificultades para rastrear el origen de ciertos cálculos.

Por este motivo, muchas empresas combinan procesos ETL y ELT según la sensibilidad de la información y el objetivo del análisis. Por ejemplo, un banco puede usar ETL para procesar los datos contables que se presentan ante entes reguladores, ya que requieren validación rigurosa y trazabilidad. En cambio, para analizar la navegación de usuarios en la aplicación móvil, puede implementar un flujo ELT, que carga los datos crudos en un *data lakehouse* y permite transformarlos de forma flexible para tareas de segmentación o personalización. Esta combinación permite trabajar con precisión en procesos críticos y con agilidad en proyectos de innovación.

CONTINUAR

Plataformas cloud para integración de datos

En la unidad anterior desarrollamos los procesos ETL y ELT, y seguramente quedó claro que integrar datos no es solo una cuestión de pasos técnicos, sino también de decisiones estratégicas. Cada enfoque responde a una lógica distinta: mientras uno transforma antes de cargar, el otro traslada esa tarea al entorno donde se almacenan los datos. Pero hay algo que los une: ambos dependen de la infraestructura sobre la que se construyen. Por eso, en este bloque vamos a detenernos en las plataformas tecnológicas que hacen posible —y sustentable— ese tipo de operaciones, especialmente en entornos donde los datos no paran de crecer.

Las plataformas *cloud* cambiaron la forma en que las organizaciones trabajan con la información. Ya no se trata únicamente de dónde guardar los datos, sino de cómo procesarlos, acceder a ellos de manera flexible y escalar sin fricciones cuando la demanda lo exige. Estas herramientas ofrecen potencia de cómputo, automatización, aislamiento entre entornos, monitoreo en tiempo real y modelos de costo

ajustables al uso. Todo esto impacta directamente en la manera en que se diseñan los flujos de integración, y por eso vale la pena entender bien cómo funcionan.

En este bloque vamos a trabajar con tres de las plataformas más utilizadas hoy en el mundo de la integración de datos en la nube: Snowflake, BigQuery y Databricks. Cada una tiene su forma particular de organizar el almacenamiento y el procesamiento, y propone modelos distintos para resolver problemas similares. Las vamos a presentar desde su arquitectura, pero también vamos a mirar cómo se usan en la práctica, qué ventajas ofrecen en distintos contextos y qué aspectos conviene tener en cuenta al elegir una u otra. La idea es que puedas incorporar criterios técnicos y operativos para tomar decisiones informadas cuando enfrentes proyectos reales.

Arquitectura y funcionamiento de Snowflake, BigQuery y Databricks

Cuando hablamos de plataformas *cloud* para la integración de datos, es importante entender que no nos referimos simplemente a lugares donde almacenar información. Se trata de soluciones completas que combinan almacenamiento, procesamiento, escalabilidad y servicios analíticos, todo en un entorno administrado. Snowflake, BigQuery y Databricks forman

parte de este conjunto de herramientas de nueva generación que permiten trabajar con grandes volúmenes de datos de forma ágil, segura y eficiente. Cada una de estas plataformas tiene una arquitectura propia, pensada para resolver necesidades específicas y con formas distintas de abordar los mismos desafíos. A continuación, vamos a analizar cada una por separado para entender cómo están construidas, cómo operan y qué posibilidades ofrecen dentro de un ecosistema moderno de integración de datos.

Snowflake: arquitectura desacoplada y escalabilidad elástica

En la unidad anterior trabajamos con los procesos ETL y ELT, y vimos cómo los datos se trasladan desde sus fuentes originales hacia entornos donde puedan organizarse y analizarse. Dependiendo del objetivo, ese entorno puede ser un *data lake*, un *data warehouse* o un *data lakehouse*. En este bloque, nos vamos a centrar en el caso de los *data warehouses*, y en particular, en una de las plataformas más utilizadas hoy para ese fin: **Snowflake**. Se trata de una solución moderna, desarrollada específicamente para la nube, que permite almacenar y consultar grandes volúmenes de datos estructurados de manera ágil, flexible y eficiente.

Ahora bien, ¿qué es un *data warehouse*? Es un sistema que organiza datos provenientes de distintas fuentes para que

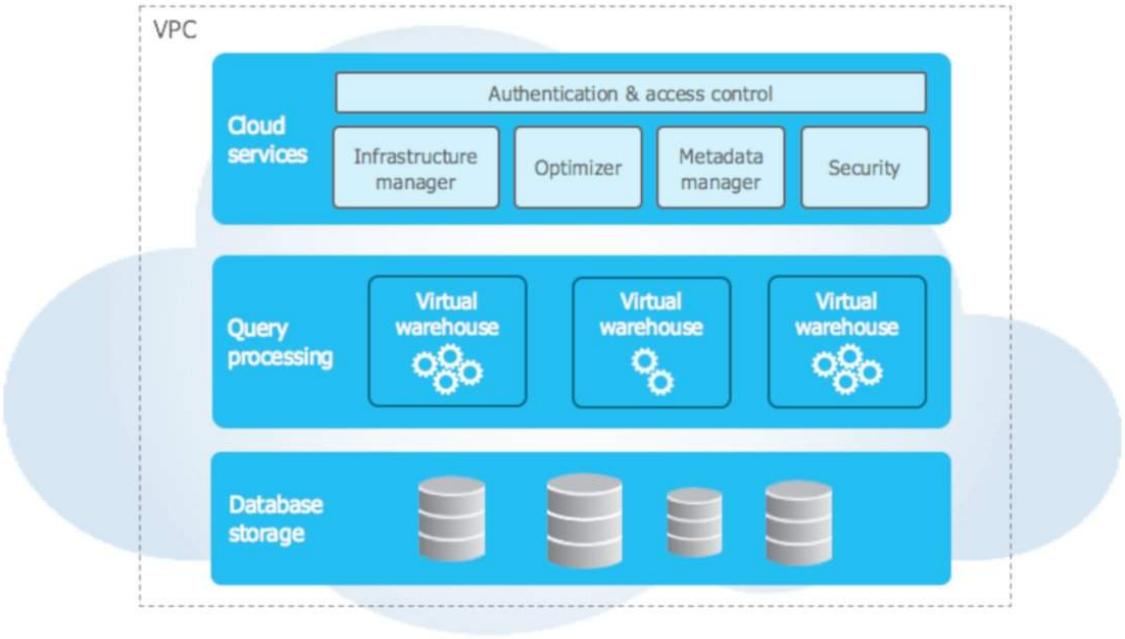
puedan ser consultados fácilmente y usados en análisis de negocio. A diferencia de un sistema operativo que gestiona transacciones del día a día, un data warehouse está pensado para responder preguntas como: ¿cuánto se vendió el último trimestre?, ¿en qué zonas crecieron las consultas?, ¿cómo cambió el comportamiento de los clientes? En otras palabras, transforma los datos dispersos en información estructurada para la toma de decisiones.

Snowflake es un ejemplo de *data warehouse cloud*, es decir, un almacén de datos diseñado desde cero para funcionar en plataformas como Amazon Web Services (AWS) o Microsoft Azure. Al ser una solución nativa en la nube, aprovecha todas las ventajas de ese entorno: escalabilidad inmediata, costos ajustados al uso, automatización de tareas técnicas y disponibilidad continua. Por eso, se vuelve una opción atractiva para integrar datos cuando se trabaja bajo el enfoque ELT, ya que permite cargar datos en crudo y transformarlos directamente dentro de la plataforma, con alto rendimiento.

Su arquitectura, como se observa en la siguiente figura, está dividida en tres capas. En la base se encuentra la capa de almacenamiento, donde los datos se guardan en formato comprimido y organizado por columnas. En el medio está la capa de procesamiento, formada por *virtual warehouses*, que son unidades de cómputo que ejecutan consultas. Finalmente, en la

parte superior están los servicios de control: autenticación, seguridad, metadatos y optimización. Esta separación permite gestionar de forma independiente cada componente, lo que se traduce en mayor flexibilidad.

Figura 5. Arquitectura de Sowflake



Fuente: Data Scientest, 2025, https://goo.su/UwzZTfw

Una de las características más valoradas de Snowflake es su capacidad para escalar de manera automática según la demanda. Supongamos que varios equipos están accediendo al sistema al mismo tiempo: finanzas generando reportes, *marketing* analizando campañas, operaciones revisando *stock*. Cada uno

puede trabajar con su propio *virtual warehouse*, sin que haya interferencias ni pérdida de rendimiento. Este modelo se conoce como *c***oncurrencia elástica,** y es clave en entornos donde muchas personas trabajan sobre los mismos datos simultáneamente.

Además, Snowflake se encarga del mantenimiento y de las optimizaciones de forma automática. No es necesario configurar servidores, ajustar parámetros técnicos ni hacer tareas de administración manual. Esto permite que los equipos se concentren en lo importante: trabajar con los datos. Si una empresa necesita actualizar sus reportes todos los días, puede automatizar la extracción y carga de datos, y aplicar las transformaciones directamente dentro de Snowflake, sin depender de servidores externos ni procesos separados.

El acceso a la plataforma es simple y versátil. Se puede utilizar desde una interfaz web, o bien conectar con otras herramientas de visualización y análisis de datos. También existen integraciones con lenguajes de programación y sistemas ETL. Esta interoperabilidad permite que Snowflake se inserte fácilmente en flujos de trabajo existentes, funcionando como el núcleo donde se almacena y se transforma la información antes de ser usada.

BigQuery: velocidad en la nube para análisis a gran escala

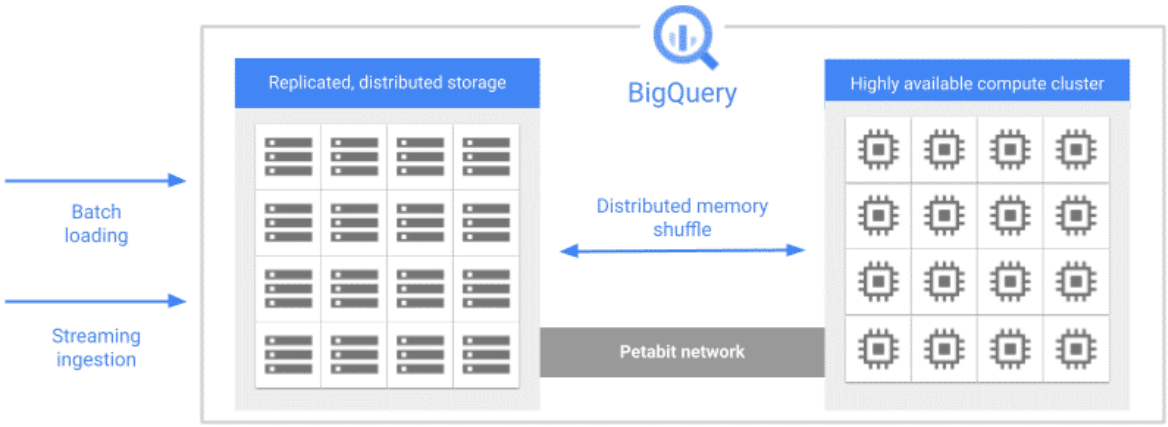
BigQuery es la solución de Google Cloud para almacenar y analizar grandes volúmenes de datos directamente en la nube. Funciona como un *data warehouse cloud*, es decir, como una plataforma que organiza datos estructurados para que puedan ser consultados con rapidez, sin necesidad de configurar servidores ni mantener infraestructura propia. Lo distintivo de BigQuery es su enfoque en el rendimiento: permite ejecutar consultas sobre millones de registros en pocos segundos, lo que la hace especialmente útil en proyectos que requieren velocidad y escalabilidad.

En muchas organizaciones, BigQuery se utiliza como el destino final de los datos integrados. Por ejemplo, una empresa puede cargar datos de ventas, comportamiento de usuarios, campañas de marketing y logística, y tener todo disponible en un solo lugar para ser analizado. Ese análisis se puede hacer directamente desde BigQuery, sin mover los datos a otra herramienta. Esto se alinea con el enfoque ELT: primero se cargan los datos tal como vienen, y luego se transforman dentro del propio entorno, aplicando reglas, cálculos o filtros según lo que necesite cada equipo.

A nivel técnico, BigQuery está diseñado con una arquitectura desacoplada: por un lado, los datos se almacenan en una infraestructura distribuida, y por otro, las consultas se procesan en un sistema de cómputo paralelo que se activa solo cuando es

necesario. Como se observa en la figura siguiente, esto permite escalar el rendimiento sin afectar la disponibilidad de los datos. La infraestructura está pensada para que el procesamiento ocurra sin interrupciones, incluso cuando varias personas están trabajando al mismo tiempo.

Figura 6. Arquitectura de BigQuery



Fuente: Google Cloud, s.f., https://goo.su/TesAj

BigQuery ofrece dos formas de incorporar datos. Una es la carga por lotes, útil para enviar archivos completos a intervalos regulares, como cada noche. La otra es la ingesta en tiempo real, donde los datos se cargan a medida que se generan. Esta opción es especialmente útil para monitorear actividades que cambian constantemente, como transacciones, sensores o clics en una

web. Ambas modalidades se pueden combinar dentro del mismo proyecto, dependiendo de los objetivos del análisis.

Cuando se consulta BigQuery, lo que ocurre en segundo plano es que la plataforma divide la tarea en múltiples partes y las procesa en paralelo. Así, una sola consulta puede ser resuelta por decenas o cientos de núcleos de procesamiento que trabajan al mismo tiempo. Esta arquitectura distribuida permite responder rápidamente incluso en bases de datos muy grandes, sin que el usuario tenga que preocuparse por cuánta capacidad está usando o cómo se debe configurar el sistema.

Uno de los aspectos más prácticos de BigQuery es que no requiere administración técnica: no hay que instalar software, ni ajustar parámetros, ni realizar tareas de mantenimiento. Todo eso lo gestiona Google automáticamente. Esto hace que los equipos de datos puedan enfocarse en construir reportes, explorar información, diseñar tableros o aplicar reglas de negocio, sin distraerse con cuestiones técnicas. Además, la plataforma se integra fácilmente con herramientas como Google Sheets, Looker Studio, Dataflow o plataformas externas de visualización y análisis.

En términos de uso, BigQuery se adapta bien a proyectos donde se necesitan consultas frecuentes, procesamiento en tiempo real y posibilidad de integrar datos de distintas fuentes. Su modelo de

cobro por consulta y su escalabilidad automática permiten ajustarse a distintos tamaños de organización, desde startups hasta empresas con grandes volúmenes de información. Es una herramienta pensada para trabajar con agilidad en entornos cambiantes, donde los datos tienen que estar disponibles y listos para analizarse en el momento que se los necesita.

Databricks: procesamiento unificado sobre data lakes

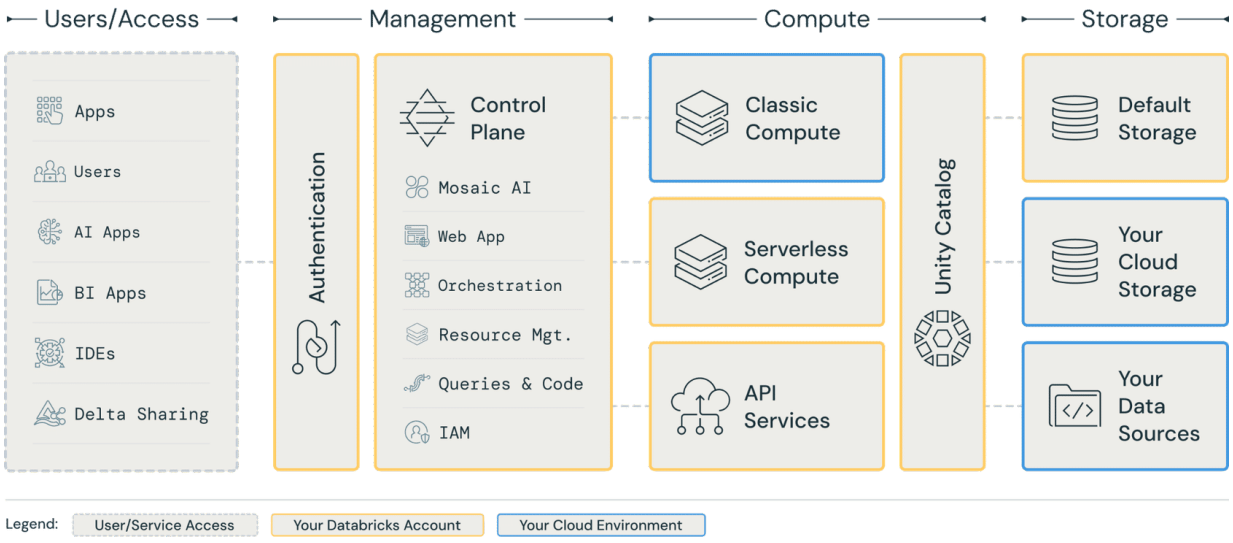
En los apartados anteriores ya vimos que, dependiendo de la arquitectura de datos que se adopte, el lugar donde se almacenan los datos puede variar: data warehouses, data lakes o soluciones híbridas como los lakehouses. En este marco, **Databricks representa un enfoque distinto al de Snowflake y BigQuery**. No es un sistema de almacenamiento, sino una **plataforma de procesamiento unificada**, capaz de trabajar directamente sobre datos almacenados en un data lake. Su propuesta se basa en integrar, transformar, analizar y gobernar datos desde una misma interfaz, sin necesidad de moverlos a otro entorno.

A nivel práctico, esto significa que Databricks **no guarda los datos**, sino que **se conecta a donde ya están almacenados** — por ejemplo, en Azure Data Lake, AWS S3 o Google Cloud Storage — y allí ejecuta los procesos necesarios. Este modelo es especialmente útil en arquitecturas ELT, donde los datos se

cargan primero en bruto y luego se transforman según las necesidades del negocio. Databricks permite aplicar esas transformaciones directamente en el data lake, sin necesidad de trasladar los datos a un data warehouse como paso intermedio.

La arquitectura de Databricks, como se muestra a continuación, está organizada en varios bloques que reflejan sus funcionalidades principales: acceso de usuarios, gestión de recursos, procesamiento y almacenamiento externo. Este diseño permite que distintos perfiles —analistas, ingenieros de datos, científicos de datos o aplicaciones de BI— puedan interactuar con los datos, ejecutar sus propias consultas o automatizar flujos de trabajo, todo dentro del mismo entorno controlado.

Figura 7. Arquitectura de Databricks



Fuente: Databricks, s.f., https://goo.su/0RKkSIn

Del lado del acceso, Databricks permite que distintos usuarios y herramientas se conecten para trabajar con los datos. Por ejemplo, un analista puede ingresar desde su entorno de trabajo habitual para consultar información; un equipo de inteligencia artificial puede utilizar datos históricos para entrenar modelos; y una herramienta de visualización puede conectarse para actualizar gráficos o reportes. También es posible compartir datos con otras plataformas externas mediante un sistema llamado Delta Sharing, que permite intercambiar conjuntos de datos de manera segura y controlada, sin tener que copiar ni mover archivos. Todas estas conexiones están gestionadas por un sistema de autenticación y permisos, que garantiza que cada usuario solo pueda acceder a los datos que le corresponden, algo especialmente importante cuando se trabaja con información sensible o confidencial.

En cuanto al procesamiento de datos, Databricks ofrece **dos formas distintas de ejecutar tareas**, según las necesidades de cada proyecto:

CLÚSTERES CLÁSICOS	COMPUTACIÓN SERVERLESS

Esta opción permite **crear entornos de trabajo temporales** con una determinada capacidad de procesamiento. Un clúster es, en términos simples, un grupo de computadoras virtuales que se activan para realizar tareas como transformar datos, entrenar modelos o generar reportes. Estos clústeres se pueden configurar según lo que se necesite (más memoria, más potencia, más tiempo de actividad) y se pueden apagar una vez que terminan su trabajo. Por ejemplo, un equipo puede levantar un clúster solo durante una hora cada mañana para procesar los datos del día anterior. Esta modalidad da control total sobre cómo se asignan los recursos, pero requiere cierta planificación.

CLÚSTERES CLÁSICOS | COMPUTACIÓN SERVERLESS

en esta modalidad, **no hace falta crear ni configurar clústeres**. Databricks se encarga automáticamente de asignar los recursos necesarios cada vez que se lanza una tarea. Es decir, si una persona o una aplicación necesita procesar datos, el sistema detecta la necesidad y **activa internamente la capacidad justa** para completar el trabajo. Una vez finalizado, los recursos se liberan solos. Este enfoque es útil cuando se quiere ganar velocidad y simplicidad, especialmente en procesos que se ejecutan con frecuencia o que deben responder de forma rápida ante distintos eventos.

Ambas formas de procesamiento pueden convivir en un mismo proyecto. Por ejemplo, una organización puede usar clústeres

clásicos para procesos programados, como el cierre mensual de ventas, y computación serverless para tareas que se activan automáticamente cuando llegan nuevos datos, como el seguimiento de envíos en tiempo real. Además, Databricks cuenta con lo que se conoce como **API Services**, que permiten que otras herramientas externas se conecten directamente a la plataforma para iniciar tareas, consultar datos o recuperar resultados, sin necesidad de hacerlo manualmente.

Por ejemplo, se puede integrar **una aplicación web** que consulta en Databricks los datos actualizados cada vez que un usuario ingresa a su perfil; o un **sistema de monitoreo de sensores** que, al detectar ciertos valores, envía una señal para que se ejecute un flujo de análisis específico. También se puede conectar **una herramienta de visualización**, como Power BI o Tableau, que consulta automáticamente los resultados de un modelo entrenado en Databricks para mostrarlos en un tablero. Estas integraciones permiten que los procesos de datos se ejecuten de forma continua y sin interrupciones, como parte de un ecosistema automatizado.

Un componente central dentro de Databricks es **Unity Catalog**, que cumple la función de organizar, controlar y proteger el uso de los datos dentro de la plataforma. Esta función se conoce como **gobernanza de datos**, y consiste en establecer reglas claras sobre **qué datos hay disponibles, quién puede acceder a**

ellos, cómo deben usarse y cómo se mantienen actualizados y confiables. En una organización donde distintos equipos trabajan con información, es fundamental evitar que cada área trabaje con versiones diferentes de los mismos datos o que alguien acceda a información que no le corresponde.

Por ejemplo, Unity Catalog permite definir que el equipo de ventas pueda consultar datos comerciales, pero no datos financieros confidenciales; o que un analista de *marketing* acceda a los reportes agregados de campañas, pero no a los registros individuales de clientes. También registra qué cambios se hacen sobre los datos y quién los realizó, lo cual permite **trazar el recorrido completo de la información**, desde su origen hasta su uso final. Esto se conoce como **trazabilidad**.

Además, Unity Catalog ayuda a evitar que se dupliquen conjuntos de datos innecesariamente. Si varios equipos necesitan usar la misma información, pueden acceder a una única versión centralizada y confiable, en lugar de generar copias separadas. Esto mejora el trabajo colaborativo, ya que todos parten del mismo punto de referencia y se minimizan los errores.

Finalmente, Databricks se conecta a distintos orígenes de datos y sistemas de almacenamiento: puede trabajar con el almacenamiento por defecto del entorno *cloud* o integrarse a estructuras ya existentes. Esto lo hace especialmente valioso en

empresas que ya tienen un *data lake* consolidado y necesitan incorporar una capa de procesamiento potente, sin migrar sus datos. De este modo, se convierte en una opción flexible y eficiente para procesar datos en bruto, integrarlos y prepararlos para ser consumidos por tableros, reportes o modelos de predicción, sin tener que replicarlos en otro sistema.

Comparativa práctica: costos, escalabilidad y rendimiento en la nube

Hasta aquí vimos tres plataformas que permiten integrar y analizar datos en la nube: Snowflake, BigQuery y Databricks. Cada una tiene su propia arquitectura, sus particularidades y su enfoque. Sin embargo, en la práctica, muchas veces se plantea una pregunta inevitable: ¿cuál conviene usar en cada caso? No se trata de elegir una como «la mejor», sino de entender en qué se diferencia cada una según las necesidades concretas del proyecto. Para eso, es útil compararlas a partir de algunos criterios que influyen directamente en las decisiones técnicas y de negocio: los costos, la escalabilidad y el rendimiento.

Tabla 1. Comparativa de Snowflake, BigQuery y Databricks según costos, escalabilidad y rendimiento

Plataforma	Costos	Escalabilidad	Rendimiento

Snowflake	Pago por uso de almacenamiento y cómputo por separado. Ideal si se planifica cuándo se usan los recursos para cargar y transformar datos.	Escala automáticamente el cómputo sin afectar el almacenamiento; se puede aumentar la capacidad según la demanda de consultas ETL o ELT.	Muy eficiente para ejecutar consultas sobre datos estructurados y mantener varios usuarios trabajando al mismo tiempo.
BigQuery	Pago por volumen de datos procesados en cada consulta. Simple, pero puede aumentar si se ejecutan muchas consultas simultáneas sobre grandes volúmenes.	Escala automáticamente según la cantidad de consultas o tamaño de datos, sin necesidad de gestionar clústeres.	Excelente velocidad para analizar grandes volúmenes de datos en la nube; ideal para procesos ELT donde se cargan datos y luego se consultan.

Databricks	Pago por el tiempo de uso de los recursos de procesamiento (*clusters*) o por tareas automáticas. Requiere gestionar cuándo y cómo se ejecutan los procesos.	Permite ajustar la capacidad de procesamiento según la cantidad de datos a transformar o consultar, asegurando que varias tareas puedan ejecutarse simultáneamente.	Muy eficiente para transformar y preparar datos dentro del entorno, especialmente en flujos ETL/ELT donde los datos llegan crudos y se transforman dentro de la plataforma.

Fuente: elaboración propia

Analicemos en detalle qué implican estas diferencias y cómo pueden impactar en el uso real.

COSTOS | ESCALABILIDAD | RENDIMIENTO

El costo de cada plataforma depende de **cómo se usan los recursos y del volumen de datos**. Snowflake cobra por separado el almacenamiento

y la potencia de cómputo. Esto permite mantener datos siempre disponibles y pagar solo cuando se realizan transformaciones. Es conveniente, por ejemplo, si una empresa carga datos de ventas durante la noche y genera reportes consolidados a primera hora, porque los recursos se utilizan solo cuando se ejecutan las tareas.

BigQuery cobra principalmente según la cantidad de datos procesados en cada consulta. Esto lo hace ideal cuando se necesitan consultas frecuentes sobre los datos cargados en bruto, como revisar la actividad de usuarios en la web durante el día o generar reportes de ventas recientes. Para que los costos no se disparen, conviene filtrar los datos y procesar solo lo necesario en cada consulta.

Databricks cobra por el tiempo de uso de los recursos de procesamiento. Esto resulta práctico cuando varios equipos trabajan sobre los mismos datos de manera simultánea. Por ejemplo, un equipo puede preparar reportes de ventas mientras otro calcula indicadores de clientes al mismo tiempo, y la plataforma ajusta automáticamente la capacidad para que se pague solo por lo que se utiliza.

COSTOS	ESCALABILIDAD	RENDIMIENTO

La escalabilidad indica cómo una plataforma puede **adaptarse al volumen de datos y al número de usuarios que trabajan con ella**, lo que afecta directamente la eficiencia de los flujos ETL y ELT.

En **Snowflake**, la escalabilidad es automática y permite separar almacenamiento y cómputo. Conviene en escenarios donde se necesita procesar grandes cargas de datos **en momentos específicos del día**, como generar los reportes consolidados de todas las sucursales al final del

mes. Aquí la plataforma aumenta la capacidad para que las consultas y transformaciones se completen rápido y luego reduce los recursos automáticamente para no generar gastos innecesarios.
BigQuery escala según la cantidad de consultas y el tamaño de los datos procesados. Esto lo hace ideal para análisis frecuentes o exploratorios que no siguen un horario fijo, como monitorear en tiempo casi real la navegación de usuarios en un portal web o consultar transacciones recientes **de manera puntual o según se necesite**, sin un plan predefinido (lo que se conoce como consultas *ad hoc*). La plataforma ajusta la capacidad automáticamente, sin que sea necesario planificar clústeres de cómputo.

En **Databricks**, la escalabilidad se gestiona a través de clústeres configurables que se pueden levantar según la demanda. Esto resulta conveniente para **procesos de integración de datos continuos o complejos**, como transformar datos de distintos sistemas de origen antes de consolidarlos en un *data lakehouse*, o aplicar reglas de limpieza y estandarización sobre datos semiestructurados que llegan en distintas frecuencias.

COSTOS	ESCALABILIDAD	RENDIMIENTO

El rendimiento indica **qué tan rápido y eficiente una plataforma puede procesar y entregar los datos** durante cargas, transformaciones o consultas, y es clave para decidir cómo implementar los flujos ETL o ELT.

En **Snowflake**, el rendimiento es especialmente alto para **consultas sobre datos estructurados**. Esto lo hace conveniente, por ejemplo, cuando se necesita generar reportes financieros que integran datos de

distintas sucursales, ya que las consultas se ejecutan rápidamente y se obtiene la información consolidada sin demoras.

BigQuery destaca por procesar grandes volúmenes de información en paralelo, ajustando automáticamente los recursos según la demanda. Esto resulta útil para **analizar registros históricos de navegación de usuarios o transacciones de clientes**, donde se requieren consultas ad hoc frecuentes y se busca obtener resultados de forma inmediata.

En **Databricks**, el rendimiento se centra en **transformar y preparar datos directamente en el entorno** antes de analizarlos. Esto conviene, por ejemplo, cuando se deben limpiar y estandarizar datos provenientes de múltiples sistemas antes de cargarlos en un *data lakehouse*, o cuando se agregan métricas derivadas a partir de distintos archivos de ventas y logística, asegurando que los datos estén listos para su análisis.

Para finalizar, y con base en lo mencionado, podemos señalar que al elegir una plataforma para la integración de datos en la nube, es importante analizar **cómo se usarán los datos, con qué frecuencia se realizarán las consultas y qué volumen de información se procesará**. Observar aspectos como la flexibilidad para transformar datos, la capacidad de escalar automáticamente y la rapidez en las consultas permite anticipar necesidades y optimizar recursos. Tomar decisiones basadas en estos criterios asegura que la plataforma elegida se adapte a los requerimientos actuales y pueda acompañar el crecimiento y la evolución de los flujos de integración de datos en la organización.

CONTINUAR

Referencias

AWS, (s.f.). *¿Qué es ETL?* https://aws.amazon.com/es/what-is/etl/

IBM, (2024). ¿Qué es ETL? https://www.ibm.com/es-es/think/topics/etl

Databricks, (s.f.). *Security & Trust Center.* https://www.databricks.com/trust/architecture

Data Scientest, (2025). *Snowflake: Cómo ha revolucionado el Data Cloud.* https://datascientest.com/es/snowflake

Google Cloud, (s.f.). *Descripción general del almacenamiento de BigQuery.* https://docs.cloud.google.com/bigquery/docs/storage_overview?hl=es-419

CONTINUAR