Big Data Aplicado

Curso de especialización en Inteligencia Artificial y Big Data.

Programación

Programación - Índice de Contenidos para el Módulo Profesional: Big Data Aplicado.

Unidad de Trabajo 1: Almacenamiento de Datos Masivos y Soluciones de Almacenamiento - Duración estimada: 12 h

- **Contenidos**:
  - **1.1.** Características y requisitos de los sistemas de almacenamiento masivo.
  - **1.2.** Tipos de almacenamiento: Relacional vs. NoSQL.
  - **1.3.** Selección de formatos de almacenamiento para Big Data.
  - **1.4.** Proceso de ingestión y limpieza de datos.
  - **1.5.** Analítica de Big Data en los ecosistemas de almacenamiento.
  - **1.6.** Big Data y Cloud.
  - **1.7.** Presentación y visualización de resultados en entornos empresariales.

Unidad de Trabajo 2: Computación Distribuida y Procesamiento de Datos - Duración estimada: 15 h

- **Contenidos**:
  - **2.1.** Introducción a la computación distribuida y paralela.
  - **2.2.** Computación paralela y distribuida con Matlab: Parallel Computing Toolbox.
  - **2.3.** Computación paralela y distribuida con Apache Spark.
  - **2.4.** Herramientas de procesamiento distribuido: MapReduce, Pig y Hive.
  - **2.5.** Automatización de trabajos con Sqoop y Oozie.
  - **2.6.** Optimización y mejora del rendimiento en entornos distribuidos.

Unidad de Trabajo 3: Integridad y Calidad de los Datos en Entornos Big Data - Duración estimada: 12 h

- **Contenidos**:
  - **3.1.** Conceptos de integridad y calidad de los datos.
  - **3.2.** Sumas de verificación y su implementación en sistemas de ficheros distribuidos.
  - **3.3.** Estrategias de mantenimiento y control de calidad en sistemas distribuidos.
  - **3.4.** Movimiento de datos entre clústeres y sistemas distribuidos.
  - **3.5.** Actualización y migración de datos en SQL-server.
  - **3.6.** Copiar o mover bases de datos entre servidores: Herramientas y procedimientos.

Unidad de Trabajo 4: Monitorización y Optimización de Clústeres - Duración estimada: 15 h

- **Contenidos**:
  - **4.1.** Herramientas de monitorización de clústeres: Interfaz web del JobTracker y Namenode.
  - **4.2.** Análisis y diagnóstico de históricos de datos.
  - **4.3.** Monitorización avanzada con Ganglia y herramientas adicionales.
  - **4.4.** Herramientas de monitorización de Hyper-V.
  - **4.5.** Apache Ambari y el ecosistema de Hadoop.
  - **4.6.** Configuración de alertas y mecanismos de respuesta automática.
  - **4.7.** Generación de informes y análisis de estabilidad de servicios.

Unidad de Trabajo 5: Aplicación de Big Data en la Inteligencia de Negocios (BI) - Duración estimada: 18 h

- **Contenidos**:
  - **5.1.** Introducción a la inteligencia de negocios (BI).
  - **5.2.** Modelos de BI basados en Big Data.
  - **5.3.** Proceso del modelo KDD: selección, limpieza y transformación de datos.
  - **5.4.** Herramientas de Business Intelligence en Big Data: SAS, IBM, Oracle, Microsoft y SAP.
  - **5.5.** Técnicas de validación de modelos.
  - **5.6.** Integración de datos estructurados y no estructurados en un modelo BI.
  - **5.7.** Implantación de modelos de BI en entornos empresariales.
  - **5.8.** Herramientas de visualización y análisis de datos (SAS Visual Analytics, SAS High-Performance Analytics, entre otros).

Unidad de Trabajo 6: Validación y Simulación de Modelos de Inteligencia de Negocios - Duración estimada: 15 h

- **Contenidos**:
  - **6.1.** Técnicas de validación de modelos BI.
  - **6.2.** Simulación de implantación de soluciones en entornos empresariales.
  - **6.3.** Evaluación del impacto de los modelos de negocio.
  - **6.4.** Análisis de resultados y mejora continua en la toma de decisiones.
  - **6.5.** Implementación de KDD con herramientas de SAS e IBM.
  - **6.6.** Ejemplos de trabajo con IBM SPSS Modeler.
  - **6.7.** Casos de uso y proyectos prácticos.

Introducción: de los datos al conocimiento

Dato: **representación sintáctica**, generalmente numérica, **que puede manejar un dispositivo electrónic**o, normalmente un ordenador. **sin significado por sí solo.**

Información es el dato interpretado, es decir, **el dato con significado**.

> →Para obtener información, ha sido necesario un proceso en el que, a **partir de un dato como elemento de entrada, se realice una interpretación de ese dato que permita obtener su significado, es decir, información a partir de él.**

La información es también el elemento de entrada y de salida en cualquier proceso de toma de decisiones

Lectura recomendada → Cuántos más datos, más conocimiento… ¿o no?

Introducción: de los datos al conocimiento

A partir de información, es posible construir conocimiento. El conocimiento es información aprendida, que se traduce a su vez en reglas, asociaciones, algoritmos, etc. que permiten resolver el proceso de toma de decisiones.

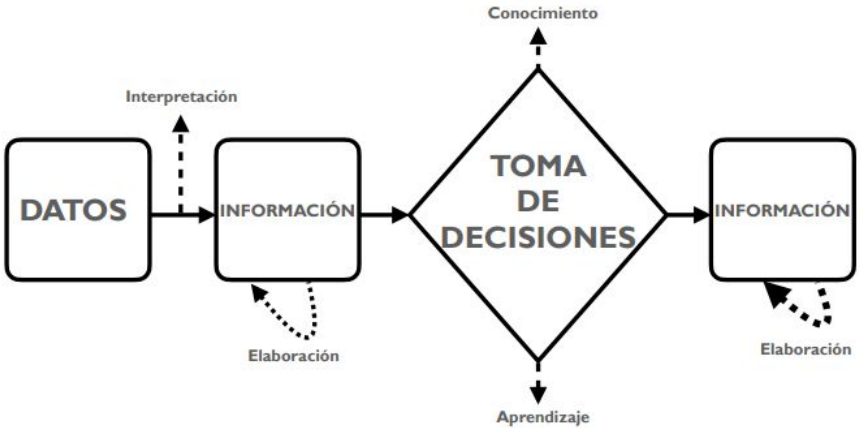


Figura 1: Relación entre datos, información y conocimiento en el proceso de toma de decisiones.

Introducción: de los datos al conocimiento

Fuente: Hey, J.: The Data, Information, Knowledge, Wisdom Chaim: The Metaphorical Link

1. El profesor corrige el examen de Pablo, que ha sacado un 3. Esta calificación, por sí sola, es simplemente un dato.
2. A continuación, el profesor calcula la calificación final de Pablo, en base a la nota del examen, sus trabajos y prácticas de laboratorio. *La nota final de Pablo es un* 4. Esto último es información.
3. ¿Ha aprobado Pablo? La información de entrada al proceso de decisión es su calificación final de 4 puntos, obtenida en el paso anterior. El conocimiento del profesor sobre el sistema de calificación le indica que una nota menor a 5 puntos se corresponde con un suspenso y, en caso contrario, con un aprobado.
4. La información de salida tras este proceso de decisión es que *Pablo está suspenso en matemáticas.*

Introducción: toma de decisiones y análisis en las empresas

¿Cómo el almacenamiento masivo afecta a la toma de decisiones y análisis en las empresas?

El almacenamiento masivo de datos **permite a las organizaciones capturar, almacenar y gestionar grandes volúmenes de datos proveniente de distintas fuentes** (transacciones, redes sociales, dispositivos IoT, etc.).

Esta capacidad de almacenar grandes cantidades de datos en diversos formatos, combinada con las herramientas de procesamiento adecuadas, **proporciona una ventaja competitiva significativa en la toma de decisiones empresariales.**

Introducción: toma de decisiones y análisis en las empresas

Aspectos clave

- **Mejora de la precisión de las decisiones:** El acceso a grandes volúmenes de datos históricos y en tiempo real permite a las empresas realizar análisis más completos y detallados. Empresa minorista → ventas históricos → datos de redes sociales puede predecir mejor las preferencias del cliente y ajustar su inventario y estrategias de marketing en consecuencia.

- **Facilita la creación de modelos predictivos:** Los modelos predictivos requieren un gran volumen de datos para identificar patrones y realizar predicciones fiables. Con la infraestructura de almacenamiento masivo adecuada (Data Lakes y Data Warehouses), los analistas pueden entrenar modelos de aprendizaje automático (ML) que ayuden a prever el comportamiento del cliente, identificar riesgos en la cadena de suministro y optimizar las operaciones.

Introducción: toma de decisiones y análisis en las empresas

Aspectos clave

- **Integración de fuentes de datos heterogéneas**: Las empresas que operan en múltiples canales necesitan gestionar datos de ventas, atención al cliente, marketing y operaciones. El almacenamiento masivo permite combinar todas estas fuentes en un único repositorio, facilitando la elaboración de informes integrados y mejorando la coherencia de los datos para la toma de decisiones.

- **Toma de decisiones en tiempo real:** Con las herramientas adecuadas, las empresas pueden utilizar grandes volúmenes de datos para tomar decisiones informadas de manera inmediata. Un ejemplo sería la detección de fraudes financieros, donde las transacciones se evalúan en tiempo real utilizando algoritmos de Big Data.

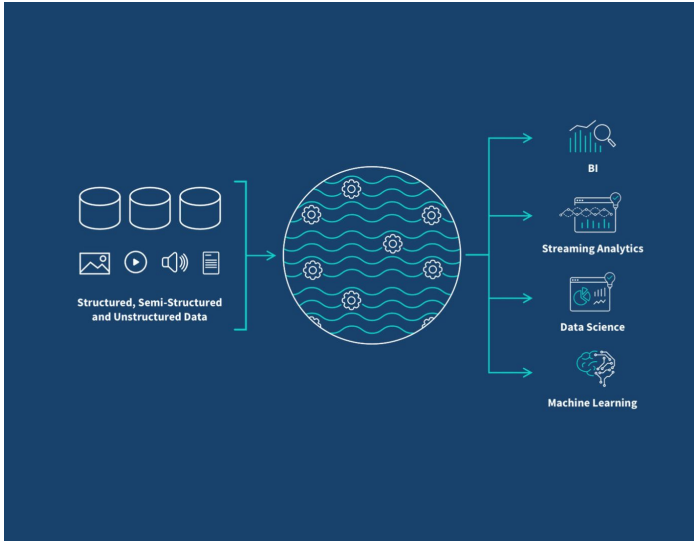
Introducción: toma de decisiones y análisis en las empresas

Implicaciones estratégicas:

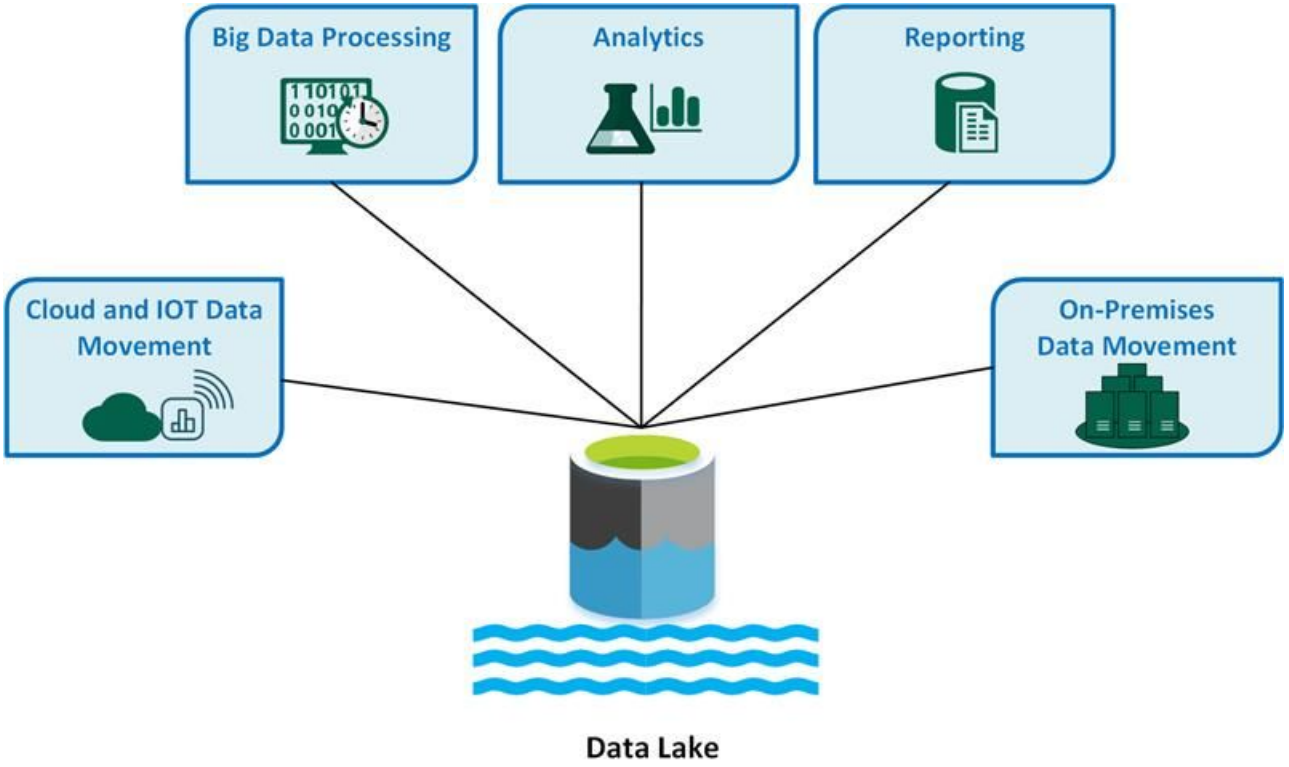
- **Reducir riesgos y costes**: Las decisiones basadas en datos masivos son menos propensas a errores porque se basan en datos más completos y actualizados. Además, las predicciones precisas permiten a las empresas optimizar sus recursos y reducir costes operativos.
- **Mejorar la experiencia del cliente**: Un análisis más detallado de los datos permite a las empresas personalizar sus ofertas y mejorar la experiencia del cliente. Por ejemplo, los algoritmos de recomendación de productos se basan en datos masivos para ofrecer sugerencias personalizadas a cada usuario.

Introducción: El rol de los Data Lakes en la transformación de datos empresariales

Los **Data Lakes** son repositorios de almacenamiento masivo diseñados para capturar y gestionar grandes volúmenes de datos en bruto y sin estructurar, provenientes de diversas fuentes (transacciones, archivos de logs, datos de sensores, imágenes, etc.). A diferencia de los Data Warehouses, los Data Lakes permiten almacenar datos en su formato original sin necesidad de transformarlos previamente. Este enfoque proporciona una gran flexibilidad para el análisis y explotación de la información.



Introducción: El rol de los Data Lakes en la transformación de datos empresariales



Introducción: El rol de los Data Lakes en la transformación de datos empresariales

Características de los Data Lakes:

- **Formato de almacenamiento flexible**: Los Data Lakes admiten todo tipo de formatos de datos, como JSON, Parquet, Avro y ORC. Esto permite a las organizaciones almacenar datos estructurados, semiestructurados y no estructurados (por ejemplo, logs de aplicaciones, datos de sensores IoT y documentos multimedia) en un único entorno.

- **Gestión de grandes volúmenes de datos a bajo coste**: La infraestructura subyacente de los Data Lakes suele basarse en sistemas de archivos distribuidos como el **Hadoop Distributed File System (HDFS) o servicios en la nube como Amazon S3.** Estas tecnologías permiten escalar la capacidad de almacenamiento a medida que aumentan las necesidades de datos, manteniendo un coste eficiente.

- **Ingestión de datos en tiempo real**: A diferencia de los Data Warehouses, que requieren un proceso de transformación previo, los Data Lakes permiten la ingesta de datos en tiempo real, lo que es esencial para aplicaciones como análisis de redes sociales y sistemas de recomendación en tiempo real.

Introducción: El rol de los Data Lakes en la transformación de datos empresariales

Beneficios del uso de Data Lakes en las empresas:

1. **Facilitan el análisis exploratorio y el descubrimiento de conocimiento**: Los Data Lakes permiten a los científicos de datos y analistas trabajar con datos sin procesar para realizar análisis exploratorios y encontrar patrones ocultos. Al tener acceso directo a la fuente original de los datos, se elimina el riesgo de pérdida de información durante los procesos de transformación.

2. **Soportan una gran variedad de casos de uso**: Los Data Lakes soportan una amplia gama de casos de uso que van desde el almacenamiento de datos en bruto hasta el procesamiento avanzado para aprendizaje automático (ML) y la integración de herramientas de Business Intelligence (BI).

3. **Permiten la democratización del acceso a los datos**: Un Data Lake bien gestionado permite a distintos usuarios dentro de la organización (analistas de negocio, desarrolladores de software, científicos de datos, etc.) acceder a los datos según sus necesidades específicas. Esto promueve la colaboración entre departamentos y facilita la creación de un entorno de análisis de datos más inclusivo.

Introducción: El rol de los Data Lakes en la transformación de datos empresariales

El Data Lake como centro de la transformación digital:

- Las empresas que implementan un enfoque de Data Lake suelen ver una mejora significativa en su capacidad para responder rápidamente a las necesidades cambiantes del negocio. Al disponer de un entorno de almacenamiento centralizado y flexible, los Data Lakes permiten experimentar con nuevos modelos de negocio basados en datos, integrar fácilmente fuentes de datos emergentes y desarrollar análisis predictivos y prescriptivos.

- **Comparación con Data Warehouses**: Mientras que los Data Warehouses están optimizados para informes y consultas predefinidas, los Data Lakes están diseñados para gestionar datos en bruto que pueden no tener un uso definido en el momento de su captura. Esto permite que los Data Lakes actúen como una fuente rica de datos para aplicaciones innovadoras como el machine learning y la inteligencia artificial, donde la calidad y cantidad de los datos es crucial.

Introducción: El rol de los Data Lakes en la transformación de datos empresariales

Implicaciones estratégicas del uso de Data Lakes:

- **Habilitación de modelos de negocio basados en datos**: Los Data Lakes permiten a las empresas almacenar grandes volúmenes de datos que pueden ser aprovechados para desarrollar nuevos productos y servicios basados en datos, como servicios de personalización avanzada y análisis predictivo.
- **Agilidad en la gestión de datos**: Las empresas pueden responder rápidamente a las nuevas oportunidades de negocio al agregar y analizar nuevas fuentes de datos sin necesidad de realizar complejos procesos de integración previos.

La carrera entre los datos y la tecnología. Obtención de información

El gran reto se basa en **extraer información a través de los datos para generar conocimiento**. Para ello será necesario que **los datos y la tecnología deben estar alineados.**

Obtener datos no ha sido siempre una tarea fácil. Esto es debido principalmente a que la gran cantidad de sensores disponibles en la actualidad, que permiten registrar magnitudes de cualquier proceso, no existía como a día de hoy.

Antes los procesos que se monitorizaban eran los procesos industriales realizados en grandes empresas. Por todos estos motivos, tradicionalmente se recurría a modelos de simulación que usaban modelos matemáticos, permitían generar datos realistas de un proceso. Los datos generados mediante simulación son conocidos como datos sintéticos mientras que los datos provenientes de las lecturas de un sensor se conocen como datos reales.

Vídeo gemelos digitales (digital twins)

La carrera entre los datos y la tecnología. Obtención de información

Para ello **es necesario contar con la tecnología necesaria para su procesamiento.** Así pues,

- **el almacenamiento se presenta como el primer problema tecnológico a resolver.**

  Se plantean **soluciones basadas en sistemas de información distribuida**. Los sistemas de información distribuida permiten adquirir espacio de almacenamiento en servidores privados, dejando la gestión de estos servidores en manos del proveedor.

- **El segundo problema tecnológico es el procesamiento de los datos almacenados.** Este aspecto cobra especial relevancia en función del caso de aplicación, pudiendo distinguirse entre:
  - **procesamiento on-line (en línea):** los datos son procesados a medida que son generados, ya que se requiere una respuesta en tiempo real.
  - **procesamiento off-line (fuera de línea):** no es necesario que los datos se procesen a medida que se generen.

La carrera entre los datos y la tecnología.

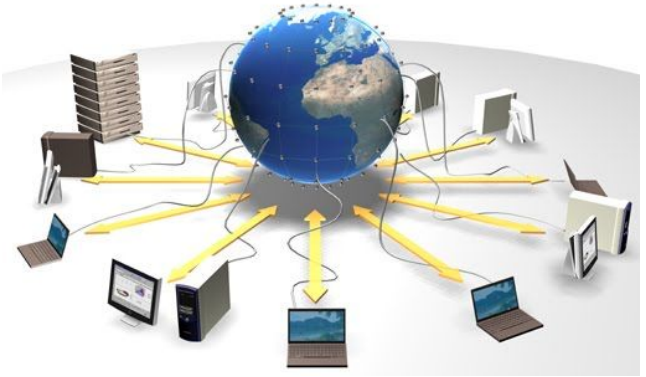
Procesamientos, tipos:

Ejemplo procesamiento online: en un sistema de control del tráfico que permite regular los semáforos en función del tráfico actual, el sistema debe regular el semáforo a medida que se van generando e interpretando los datos del tráfico en un instante de tiempo dado.

Ejemplo procesamiento off-line: en un sistema de detección del fraude bancario, comprobar si un cliente ha realizado algún movimiento fraudulento es una tarea que puede llevarse a cabo off-line, por ejemplo, haciendo un análisis de los movimientos del cliente en un momento dado, sin tener por qué diagnosticar cada movimiento que este va realizando.

La carrera entre los datos y la tecnología

La computación distribuida, en donde múltiples máquinas realizan el procesamiento **optimizando el rendimiento o la computación en la nube**, que permite adquirir recursos de procesamiento al igual que se puede adquirir espacio de almacenamiento, son dos soluciones al problema del procesamiento.



Otras alternativas son:

- **la programación paralela:** aprovechar el paralelismo de múltiples hilos de ejecución dentro de un procesador
- **la programación multi-procesador:** realizar el procesamiento dividiéndolo en múltiples hilos en diferentes procesadores.

La carrera entre los datos y la tecnología.

Data Warehouses y **Data Lakes**

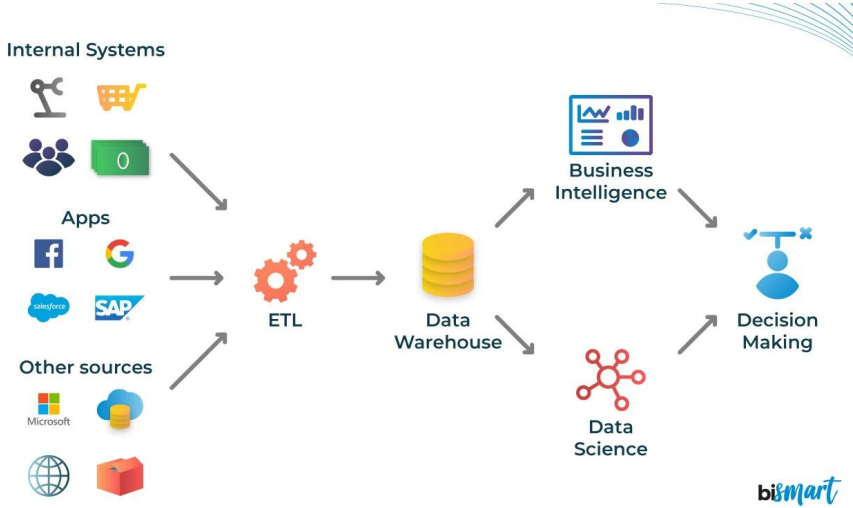
Data Warehouses: Un **Data Warehouse** es un repositorio centralizado que almacena datos estructurados y procesados, organizados con un esquema definido. Los Data Warehouses son utilizados principalmente para la generación de informes y el análisis de datos históricos en aplicaciones de inteligencia de negocios (BI).

- **Características principales:**
  - **Estructura definida**: Los Data Warehouses están organizados en tablas y esquemas bien definidos, donde cada dato tiene una posición y un significado específicos.
  - **Almacenamiento optimizado para consultas**: La estructura de los Data Warehouses está diseñada para responder a consultas SQL complejas de manera eficiente.
  - **Transformación previa**: Los datos que ingresan a un Data Warehouse pasan por procesos ETL (Extracción, Transformación y Carga), que limpian, integran y transforman los datos en un formato coherente.
  - **Uso típico**: Se utilizan para informes de rendimiento empresarial, análisis financiero y cuadros de mando.

La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes

- **Ventajas de los Data Warehouses:**
  - Excelente rendimiento para análisis de datos históricos.
  - Alta fiabilidad en la generación de informes.
  - Seguridad y control de acceso robustos.
- **Limitaciones:**
  - No es adecuado para almacenar datos n estructurados.
  - Los procesos de ETL pueden ser lentos y costosos.
  - Menor flexibilidad para adaptarse a cambios en los datos.



La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes

Data Lakes: Un **Data Lake** es un almacén centralizado que permite almacenar grandes volúmenes de datos en su formato original, sin procesar. Los Data Lakes admiten datos estructurados, semiestructurados y no estructurados, como logs, archivos de audio, video y datos de sensores.

- **Características principales:**
  - **Formato de almacenamiento flexible**: Los Data Lakes pueden almacenar datos en múltiples formatos (JSON, CSV, imágenes, vídeos, etc.).
  - **Sin esquema predefinido**: Los datos no requieren un esquema predefinido. Esto permite almacenar datos sin transformación previa, lo cual es útil para análisis exploratorios.
  - **Ingestión rápida**: Los datos se pueden almacenar inmediatamente sin pasar por un proceso ETL, permitiendo la ingesta en tiempo real.
  - **Uso típico**: Se utilizan para análisis exploratorios, machine learning y procesamiento avanzado.

La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes

Data Lakes:

- **Ventajas de los Data Lakes:**
- Alta flexibilidad para trabajar con cualquier tipo de dato.
- Permite realizar análisis avanzados y machine learning.
- Escalabilidad a bajo coste.

- **Limitaciones:**
  - Puede convertirse en un “Data Swamp” si no se gestionan adecuadamente, es decir, un repositorio desorganizado donde es difícil encontrar valor.
  - Requiere procesos de gobernanza de datos y herramientas de catalogación para mantener la calidad y accesibilidad.

La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes

Comparativa

Aspecto	Data Warehouse	Data Lake
Tipo de datos	Estructurados	Estructurados, semiestructurados y no estructurados
Proceso de ingestión	ETL (datos transformados antes de almacenar)	ELT (datos transformados después de almacenar)
Arquitectura	Modelo relacional (tablas y esquemas)	Sin estructura fija
Costo	Mayor coste de almacenamiento	Menor coste debido a almacenamiento en bruto
Velocidad	Óptimo para consultas y reportes rápidos	Más lento en consultas de datos masivos sin índice
Casos de uso	BI, informes de ventas y rendimiento	Análisis avanzado, machine learning, data mining

La carrera entre los datos y la tecnología.

Data Warehouses y **Data Lakes**

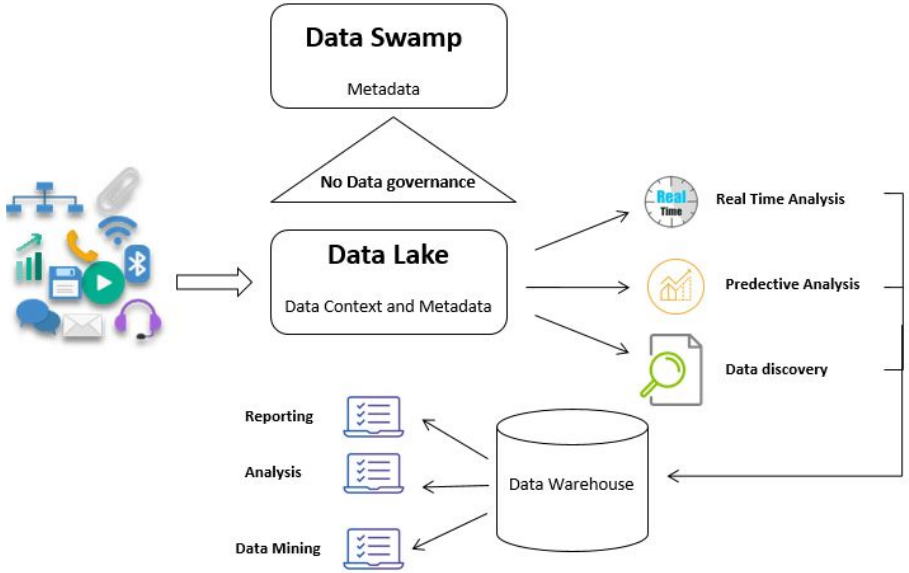
¿Cuándo usar Data Warehouses y cuándo Data Lakes?

- Utiliza **Data Warehouses** cuando:
  - Se necesiten análisis detallados y rápidos de datos históricos.
  - El esquema de los datos sea estable y esté bien definido.
  - Se requiera un control riguroso del acceso a los datos.
- Utiliza **Data Lakes** cuando:
  - Se necesiten datos para análisis exploratorio y desarrollo de modelos predictivos.
  - Los datos no estén estructurados o provengan de múltiples fuentes.
  - Se quiera evitar el alto coste de transformación de datos antes de almacenarlos.

La carrera entre los datos y la tecnología.

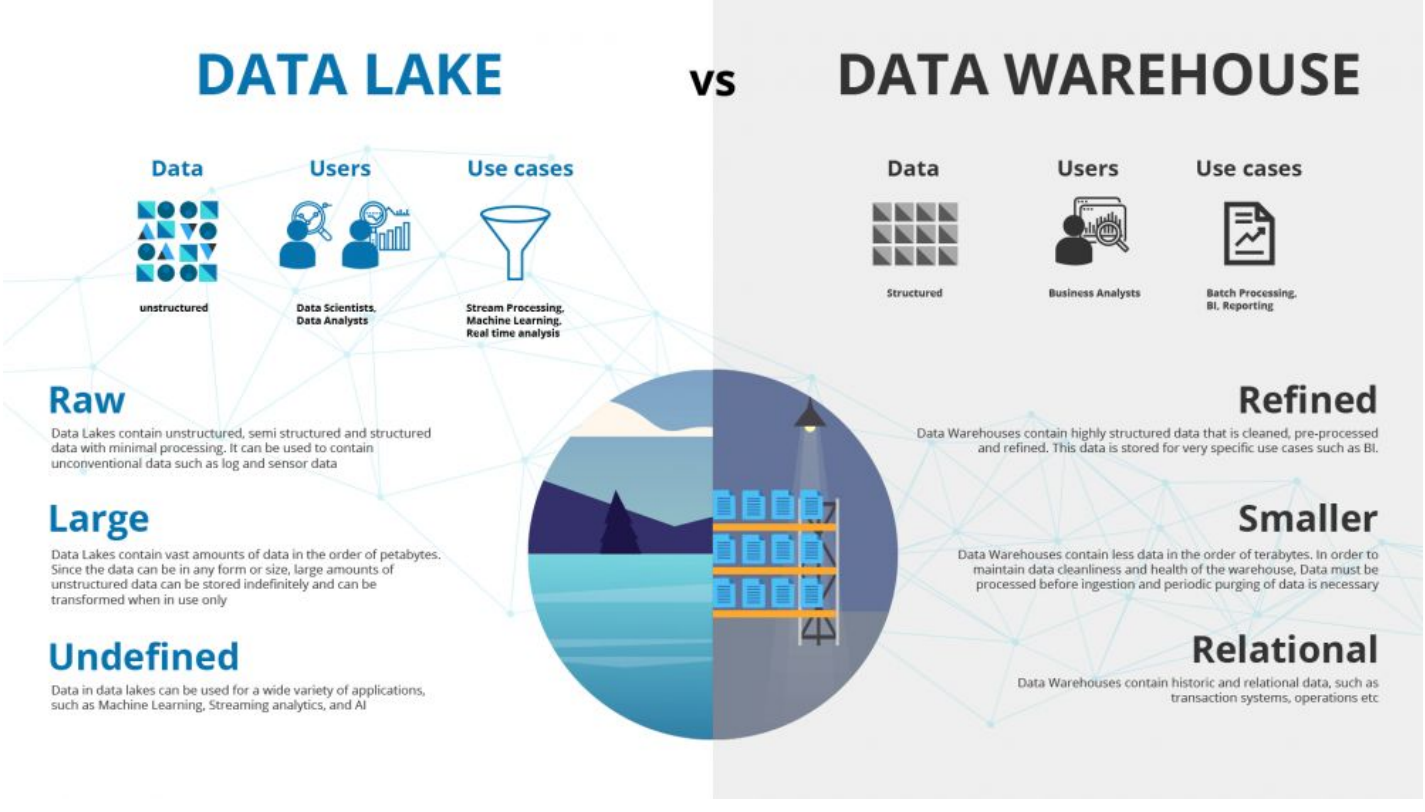
Data Warehouses y **Data Lakes**

Integración de Data Lakes y Data Warehouses: Cada vez más, las organizaciones combinan Data Warehouses y Data Lakes para obtener lo mejor de ambos mundos. Se pueden utilizar los **Data Lakes como repositorios primarios** para almacenar grandes volúmenes de datos en bruto, mientras que los Data Warehouses se utilizan como **repositorios secundarios** para transformar estos datos y preparar informes optimizados.



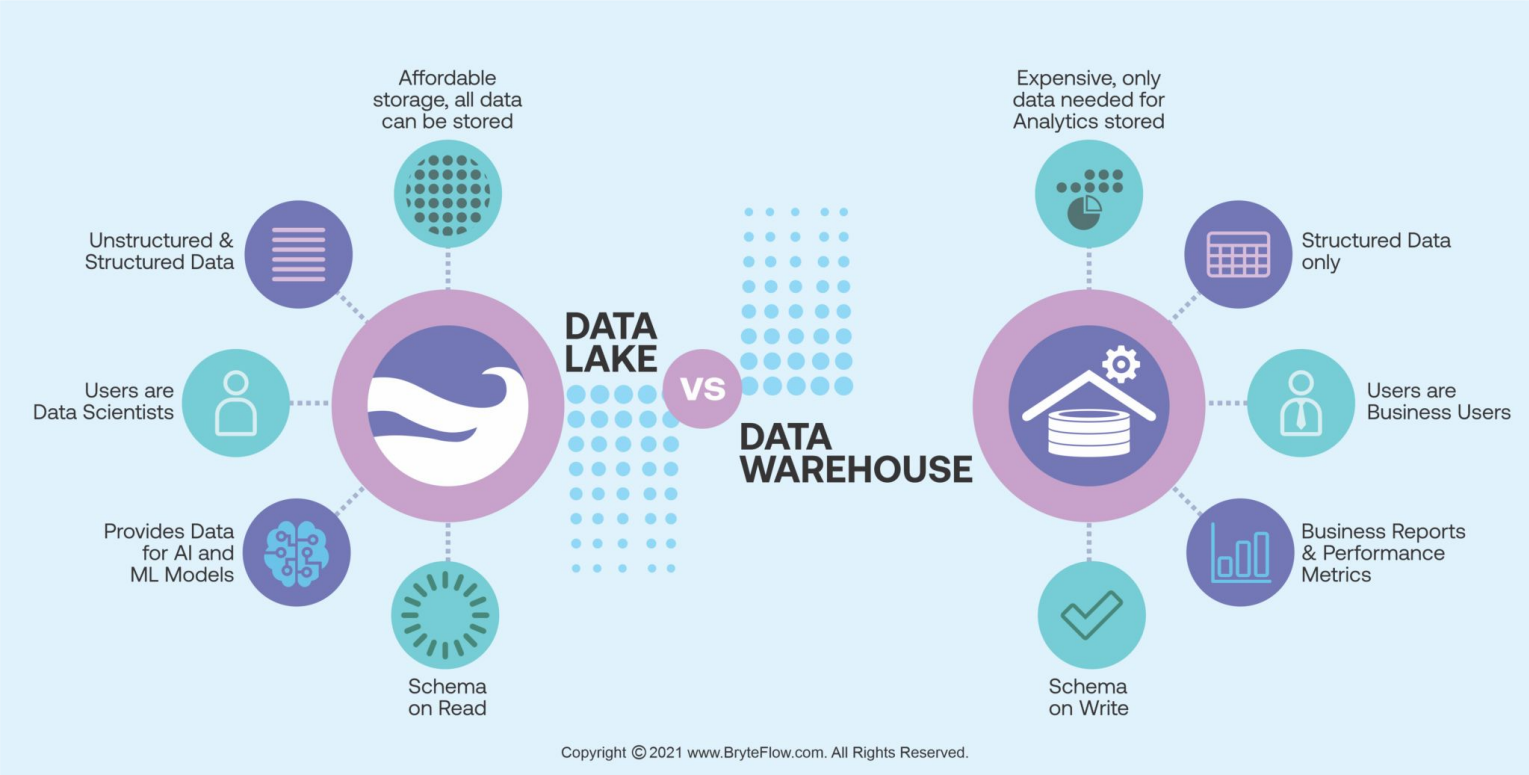
La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes



La carrera entre los datos y la tecnología.

Data Warehouses y Data Lakes



La carrera entre los datos y la tecnología.

Cloud Computing y su aplicación en Big Data

El uso de **Cloud Computing** ha transformado la forma en que las empresas gestionan sus datos masivos, proporcionando la infraestructura necesaria para escalar, procesar y analizar grandes volúmenes de información sin la necesidad de implementar y gestionar infraestructura física.

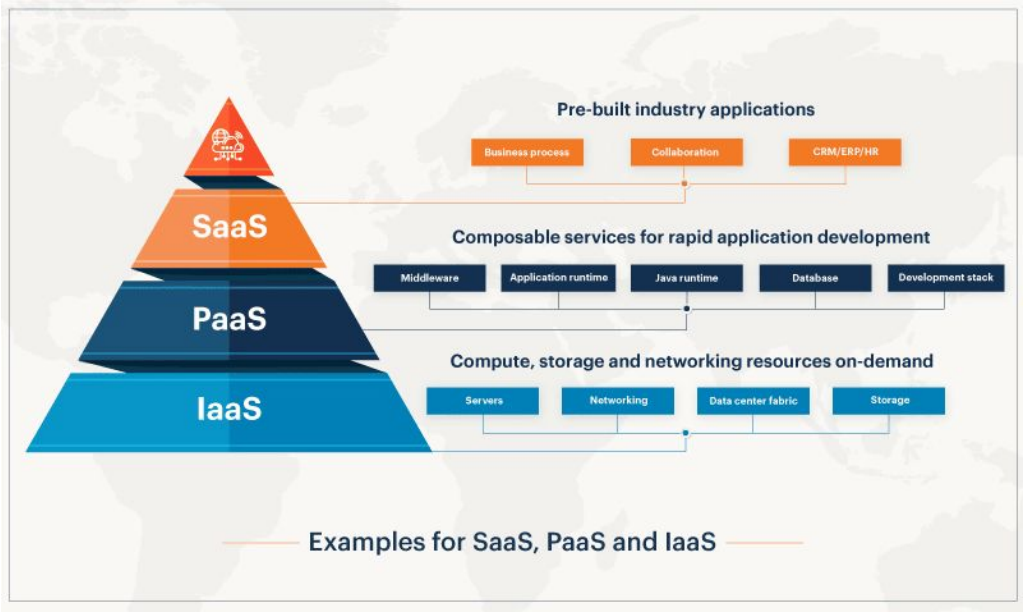
El Cloud Computing permite a las organizaciones acceder a recursos de computación (almacenamiento, servidores, bases de datos, etc.) a través de internet, pagando solo por lo que usan. Los servicios de computación en la nube se dividen en tres categorías principales:

- **Infraestructura como Servicio (IaaS)**
- **Plataforma como Servicio (PaaS)**
- **Software como Servicio (SaaS)**

La carrera entre los datos y la tecnología.

Cloud Computing y su aplicación en Big Data

- **Infraestructura como Servicio (IaaS)**: Proporciona máquinas virtuales y recursos de almacenamiento para construir y ejecutar aplicaciones en la nube.
- **Plataforma como Servicio (PaaS)**: Ofrece plataformas completas para el desarrollo y despliegue de aplicaciones, gestionando la infraestructura subyacente.
- **Software como Servicio (SaaS)**: Permite acceder a aplicaciones completas a través de la web, como CRM y suites de productividad.



La carrera entre los datos y la tecnología.

Beneficios del Cloud Computing en Big Data:

1. **Escalabilidad ilimitada**: Los entornos en la nube permiten a las empresas aumentar o reducir rápidamente la capacidad de almacenamiento y procesamiento según la demanda.
2. **Menor coste operativo**: Se eliminan los gastos de inversión en hardware físico, ya que los costos se basan en el consumo real de recursos (modelo de pago por uso).
3. **Disponibilidad y redundancia**: Los servicios en la nube ofrecen alta disponibilidad, redundancia de datos y recuperación ante desastres.
4. **Ingesta y procesamiento de datos en tiempo real**: Con herramientas como **AWS Kinesis** o **Azure Stream Analytics**, las empresas pueden ingestar y procesar datos en tiempo real.

La carrera entre los datos y la tecnología.

Principales proveedores de Cloud Computing en Big Data:

1. **Amazon Web Services (AWS)**:
   - **Amazon Redshift**: Data Warehouse basado en la nube.
   - **Amazon S3**: Almacenamiento escalable en Data Lakes.
   - **Amazon EMR**: Plataforma gestionada para procesamiento Big Data.
2. **Microsoft Azure**:
   - **Azure Synapse Analytics**: Integra Data Warehousing y análisis de Big Data.
   - **Azure Data Lake Storage**: Repositorio escalable para almacenar datos sin procesar.
   - **Azure HDInsight**: Servicio para análisis Big Data basado en Hadoop.
3. **Google Cloud Platform (GCP)**:
   - **BigQuery**: Almacenamiento y análisis de grandes volúmenes de datos.
   - **Cloud Dataflow**: Herramienta de procesamiento en tiempo real y por lotes.

La carrera entre los datos y la tecnología.

Desafíos del Cloud Computing en Big Data:

1. **Costos ocultos**: Si no se gestionan adecuadamente, los costos de la nube pueden escalar rápidamente.
2. **Seguridad y cumplimiento**: Es crucial garantizar la seguridad de los datos y el cumplimiento de regulaciones (GDPR, CCPA).
3. **Latencia y rendimiento**: La transferencia de grandes volúmenes de datos a través de la red puede afectar el rendimiento de las aplicaciones.

Estrategias de implementación:

- **Híbrido**: Una combinación de infraestructura en la nube y on-premise para mantener el control de los datos sensibles.
- **Multi-Cloud**: Uso de múltiples proveedores para evitar la dependencia de un único servicio y maximizar las capacidades.

Los datos: de ayer y de hoy

La tecnología ha ido evolucionando para dar respuesta a la ingente cantidad de datos que ha comenzado a generarse. **Esta evolución, o revolución,** no está únicamente relacionada con la **cantidad de datos** (como se expuso en el anterior apartado) sino también con el **tipo y el formato de los mismos**.

En el pasado los formatos de archivos que se manejaban solían ser formatos de hojas de cálculo (.xlsx, .ods, .numbers etc) o ficheros separados por comas (.csv). Muy pocos eran los procesos en los que se trabajaba con otros tipos de datos como **texto, imágenes, audio e incluso vídeos**, ya que los formatos de estos tipos de datos eran limitados hace unos años, su procesamiento más complejo y la tecnología para ello aún en desarrollo.

Los datos: de ayer y de hoy

- **En cuanto al texto,**las técnicas de inteligencia artificial y procesamiento del lenguaje natural **hacen posible la extracción de conocimiento a partir de grandes volúmenes de textos,** que pueden provenir de páginas web, archivos .pdf, redes sociales, etc.
- El **desarrollo de hardware** con mejores prestaciones y los nuevos modelos de programación **permiten procesar en la actualidad grandes cantidades de imágenes, audios y vídeos** con una gran variedad de técnicas de inteligencia artificial en tiempos razonables.
- **Aparición de nuevos tipos y formatos de datos como los generados a partir de grafos.** Estos datos se corresponden, por ejemplo, con datos geográficos obtenidos a partir de mapas como los generados en aplicaciones como Google Maps u Open Street Maps o datos de seguimiento y actividad en redes sociales de gran valor en campañas publicitarias entre otros muchos

Los datos: de ayer y de hoy

- El **desarrollo de bases de datos NoSQL más flexibles y escalables horizontalmente**, que permiten almacenar y consultar eficientemente los diversos tipos de datos generados. Por ejemplo MongoDB, Cassandra, etc.
- La **computación en la nube**, que facilita el procesamiento masivo de datos sin necesidad de infraestructura propia. Servicios como AWS, Azure, GCP proporcionan recursos bajo demanda para ejecutar trabajos de big data.
- La **aplicación de técnicas de aprendizaje automático y deep learning** sobre conjuntos de datos masivos, permitiendo entrenar modelos más precisos para tareas como clasificación, predicción, reconocimiento de patrones, etc.
- El concepto de **data lakes, repositorios centralizados** que almacenan y organizan grandes cantidades de datos sin procesar, para su posterior análisis.

Big Data

A diario se generan enormes cantidades de datos, del orden de petabytes. Se estima que **el 90% de los datos disponibles en el mundo ha sido generado en los últimos años.**

La capacidad de enviar y recibir datos e información a gran velocidad, así como la capacidad de almacenar tal cantidad de datos y procesarlos en tiempo real. **Así pues, la gran cantidad de datos disponibles junto con las herramientas, tanto hardware como software, que existen a disposición para analizarlos se conoce como big data.**

Big Data

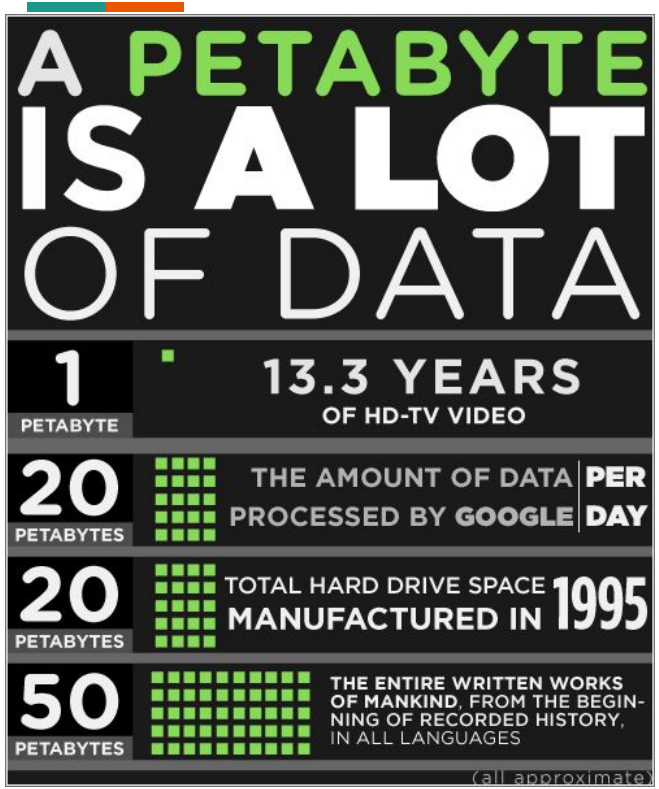

Unidades de almacenamiento
IONOS
Bit
Nibble
= 4 Bit
= 4 Bit
Byte
= 2 Nibble
= 2 Nibble
Kilobyte
= 1.000 Byte
= 1.024 Byte
Megabyte
= 1.000 Kilobyte
= 1.024 Kilobyte
Gigabyte
= 1.000 Megabyte
= 1.024 Megabyte
Terabyte
= 1.000 Gigabyte
= 1.024 Gigabyte
Petabyte
= 1.000 Terabyte
= 1.024 Terabyte
Exabyte
= 1.000 Petabyte
= 1.024 Petabyte
Zettabyte
= 1.000 Exabyte
= 1.024 Exabyte
Yottabyte
= 1.000 Zettabyte
= 1.024 Zettabyte
Brontobyte
= 1.000 Yottabyte
= 1.024 Yottabyte
Decimal
Binario

Big Data

Ejemplos prácticos de unidades de almacenamiento:

- **Byte:** La unidad más pequeña, difícil de visualizar por sí sola. Imagina que un byte puede almacenar una letra del alfabeto o un número pequeño.
- **Kilobyte (KB):** Un texto corto, como un párrafo o un tweet, puede ocupar aproximadamente 1 KB.
- **Megabyte (MB):** Una canción en formato MP3 suele tener un tamaño de 3-5 MB. **Una foto de alta resolución puede ocupar entre 2 y 5 MB.**
- **Gigabyte (GB):** Una película en alta definición (HD) puede ocupar entre 4 y 8 GB. Un juego de video para consola puede ocupar varios gigabytes.
- **Terabyte (TB):** Un disco duro externo típico tiene una capacidad de 1 TB o más. Puede almacenar miles de fotos, cientos de películas o decenas de juegos.
- **Petabyte (PB):** Un centro de datos grande puede almacenar petabytes de información. **Esto equivale a millones de libros o miles de horas de video de alta calidad.**
- **Exabyte (EB): La cantidad de información generada en el mundo cada dos días se estima en varios exabytes.**
- **Zettabyte (ZB):** Una cantidad de datos difícil de imaginar para una persona individual. Se utiliza para medir la capacidad de almacenamiento de los centros de datos más grandes del mundo.
- **Yottabyte (YB):** Una unidad de almacenamiento extremadamente grande, utilizada para medir la capacidad de almacenamiento total de internet.
- **Brontobyte: La unidad más grande de la lista, difícil de visualizar en un contexto real.**

Big Data


A PETABYTE
IS A LOT
OF DATA
1
PETABYTE
13.3 YEARS
OF HD-TV VIDEO
20
PETABYTES
THE AMOUNT OF DATA PROCESSED BY GOOGLE PER DAY
20
PETABYTES
TOTAL HARD DRIVE SPACE MANUFACTURED IN 1995
50
PETABYTES
THE ENTIRE WRITTEN WORKS OF MANKIND, FROM THE BEGINNING OF RECORDED HISTORY, IN ALL LANGUAGES
(all approximate)

Big Data

Actividad:

- Lee el siguiente artículo.

  HOW MUCH DATA IS GENERATED EVERY DAY IN 2024? (NEW STATS)

  Read more at EarthWeb: How Much Data Is Generated Every Day in 2024? (NEW Stats) https://earthweb.com/how-much-data-is-created-every-day/

- Investiga y reflexiona sobre la importancia de los datos en la actualidad, y cómo su creciente generación afecta a la sociedad y a las empresas

Big Data - HOW MUCH DATA IS GENERATED EVERY DAY IN 2024? (NEW STATS)

Reflexiones:

- A medida que aumenta su volumen, también lo hace su **valor estratégico.**
- Los datos permiten a las empresas entender mejor el comportamiento de sus clientes, **optimizar procesos operativos y tomar decisiones basadas en información** en lugar de intuición.
- Analizar grandes volúmenes de información en tiempo real **permite detectar tendencias emergentes, prever problemas potenciales y reaccionar con mayor agilidad ante cambios en el entorno.**
- Los sistemas de transporte inteligente, l**a personalización de servicios y la automatización** de tareas dependen de la recopilación y procesamiento de datos
- Este crecimiento desmedido también plantea desafíos importantes en términos de **privacidad y seguridad.**

Big Data - HOW MUCH DATA IS GENERATED EVERY DAY IN 2024? (NEW STATS)

Impacto en las Empresas

Para las empresas, **los datos** no son sólo un subproducto de la actividad digital, sino **un recurso estratégico que puede definir su competitividad en el mercado**. La capacidad de manejar y extraer valor de los datos es cada vez más un diferenciador clave. Algunas de las implicaciones de esta explosión de datos incluyen:

1. **Toma de Decisiones Basada en Datos**: Las empresas pueden tomar decisiones informadas basadas en análisis detallados y en tiempo real de grandes volúmenes de información.
2. **Personalización del Servicio**: Los datos permiten a las organizaciones personalizar sus productos y servicios de manera más efectiva, creando experiencias más satisfactorias para los clientes.
3. **Innovación y Nuevos Modelos de Negocio**: La disponibilidad de datos impulsa la innovación, permitiendo a las empresas identificar nuevas oportunidades y desarrollar modelos de negocio basados en análisis predictivo e inteligencia artificial.
4. **Cumplimiento Normativo**: A medida que aumentan los volúmenes de datos, también lo hacen las regulaciones destinadas a proteger la privacidad de los usuarios y la seguridad de la información.

Big Data - HOW MUCH DATA IS GENERATED EVERY DAY IN 2024? (NEW STATS)

Retos y Consideraciones Futuras

El principal reto que enfrentan las empresas y gobiernos es **gestionar y proteger** adecuadamente este vasto universo de información. La evolución hacia sistemas de seguridad basados en el modelo de **Confianza Cero**, que se centra en autenticar continuamente a los usuarios y proteger los datos independientemente de su ubicación, es un paso crucial para enfrentar los riesgos asociados. Además, l**a automatización de procesos de gobernanza de datos y la aplicación de inteligencia artificial para la detección de amenazas serán esenciales para mantener la integridad y privacidad de la información.**

Big Data

Actividad, continuación.

- Según el informe "Data Never Sleeps 9.0", elaborado por Domo, en 2023 se generan más de 2,5 quintillones de bytes de datos cada día. Esto equivale a unos 73 zettabytes de datos al año.

- El crecimiento de los datos se debe a varios factores, entre los que se incluyen:

    - El aumento del uso de dispositivos conectados, como teléfonos inteligentes, tablets y ordenadores.
    - La proliferación de las redes sociales y las aplicaciones de mensajería instantánea.
    - La creciente popularidad de la nube y el almacenamiento en línea.

Big Data

Actividad, continuación.

Año	Cantidad de datos generados (bytes)	Tráfico de Internet (%)	Redes sociales (terabytes)
2020	2,5 quintillones	90%	500
2025	463 exabytes	90%	2,5 petabytes

- Año: Año en el que se generaron los datos.
- Cantidad de datos generados (bytes): Cantidad total de datos generados en ese año.
- Tráfico de Internet (%): Porcentaje de los datos generados que se originaron en el tráfico de Internet.
- Redes sociales (terabytes): Cantidad de datos generados por las redes sociales en ese día.

Big Data

Actividad, continuación.

- El aumento de los datos tiene un impacto significativo en la sociedad, ya que abre nuevas oportunidades para la innovación y el desarrollo. Sin embargo, también plantea desafíos, como la necesidad de desarrollar nuevas tecnologías para almacenar y procesar grandes cantidades de datos.

Especifidad	Datos
Cantidad de datos generados cada día	2,5 quintillones de bytes
Equivalencia en bytes	73 zettabytes
Factores que impulsan el crecimiento de los datos	Uso de dispositivos conectados, redes sociales, nube
Impacto de los datos	Oportunidades e innovaciones, desafíos

Big Data

Formatos de Datos en Big Data: Análisis de pros y contras

El **tipo de formato de datos** que se elige para almacenar la información es crucial, ya que impacta directamente en la **eficiencia de almacenamiento**, **rendimiento** de procesamiento, **compatibilidad** con herramientas de análisis y **costos** de almacenamiento. A continuación, se presentan algunos de los **formatos de datos más utilizados en entornos de Big Data** (comparativa):

Formato	Tipo	Eficiencia de Almacenamiento	Facilidad de Uso	Rendimiento	Casos de Uso
CSV	Texto plano	Baja	Alta	Moderado	Intercambio de datos, informes
JSON	Texto plano	Baja	Moderada	Moderado	APIs, datos semiestructurados
Parquet	Columnar	Alta	Moderada	Alta (lectura)	Big Data Analytics, Data Warehouses
ORC	Columnar	Muy alta	Moderada	Muy alta	Consultas analíticas masivas
Avro	Binario con esquema	Moderada	Baja	Alta (escritura)	Flujos de datos, transmisión

Big Data

Recomendación de uso:

- **Parquet** u **ORC** para grandes volúmenes de datos cuando se requiere almacenamiento optimizado y consultas rápidas.
- **CSV** para almacenamiento temporal o cuando se necesita un formato simple para compartir datos.
- **Avro** para transmisión de datos en entornos de streaming (por ejemplo, con Apache Kafka).
- **JSON** cuando se trabaja con datos semiestructurados que necesitan flexibilidad.

Big Data. Casos de uso de almacenamiento distribuido

El almacenamiento distribuido permite gestionar grandes volúmenes de datos en múltiples nodos, garantizando alta disponibilidad, tolerancia a fallos y escalabilidad.

Casos de uso de almacenamiento distribuido en diferentes entornos:

Almacenamiento y Procesamiento de Datos Masivos en Empresas de Comercio Electrónico

- **Ejemplo**: Una tienda online gestiona millones de transacciones al día, junto con datos de clics en el sitio web, actividad de usuarios y opiniones de productos. Para ello deberá implementar un sistema de almacenamiento distribuido con **Hadoop HDFS** y **Apache Hive** para organizar los datos de ventas y comportamiento de usuarios en tablas estructuradas. Utilizar **Spark** para procesar y analizar los datos a gran escala.
- **Resultado**: La empresa puede identificar patrones de compra, realizar análisis de tendencias y mejorar la personalización de ofertas.

Big Data. Casos de uso de almacenamiento distribuido

Análisis de Logs y Monitorización en Sistemas TI

- **Ejemplo**: Un proveedor de servicios en la nube gestiona grandes volúmenes de logs generados por miles de servidores y dispositivos. Para ello almacenará los logs en bruto en un **Data Lake** basado en **Amazon S3** y utilizar **Amazon EMR** para analizar los patrones de acceso, identificar anomalías y realizar diagnósticos.
- **Resultado**: Mejora en la monitorización de infraestructura, detección de fallos y reducción de tiempos de inactividad.

Almacenamiento y Procesamiento de Datos IoT en Tiempo Real

- **Ejemplo** Una empresa de manufactura recopila datos de sensores en tiempo real de múltiples fábricas para monitorear el rendimiento de la maquinaria. Para ello deberá implementar un almacenamiento distribuido con **Hadoop HDFS** y **Apache Kafka** para la ingesta de datos en tiempo real. Utilizar **Apache Flink** para procesar los flujos de datos y realizar cálculos en tiempo real.
- **Resultado**: Identificación temprana de problemas en las máquinas, reducción de tiempo de inactividad y optimización de la producción.

Cluster de computadoras/ordenadores

Los clusters son **conjuntos de máquinas interconectadas que trabajan como una única unidad para resolver cargas de trabajo de manera conjunta.** Estos ofrecen varias ventajas, como **alto rendimiento, alta disponibilidad, equilibrio de carga, escalabilidad y tolerancia a fallos en sistemas que manejan grandes volúmenes de datos.**

Cluster de computadoras/ordenadores

Cómo los clústeres soportan la escalabilidad y procesamiento masivo

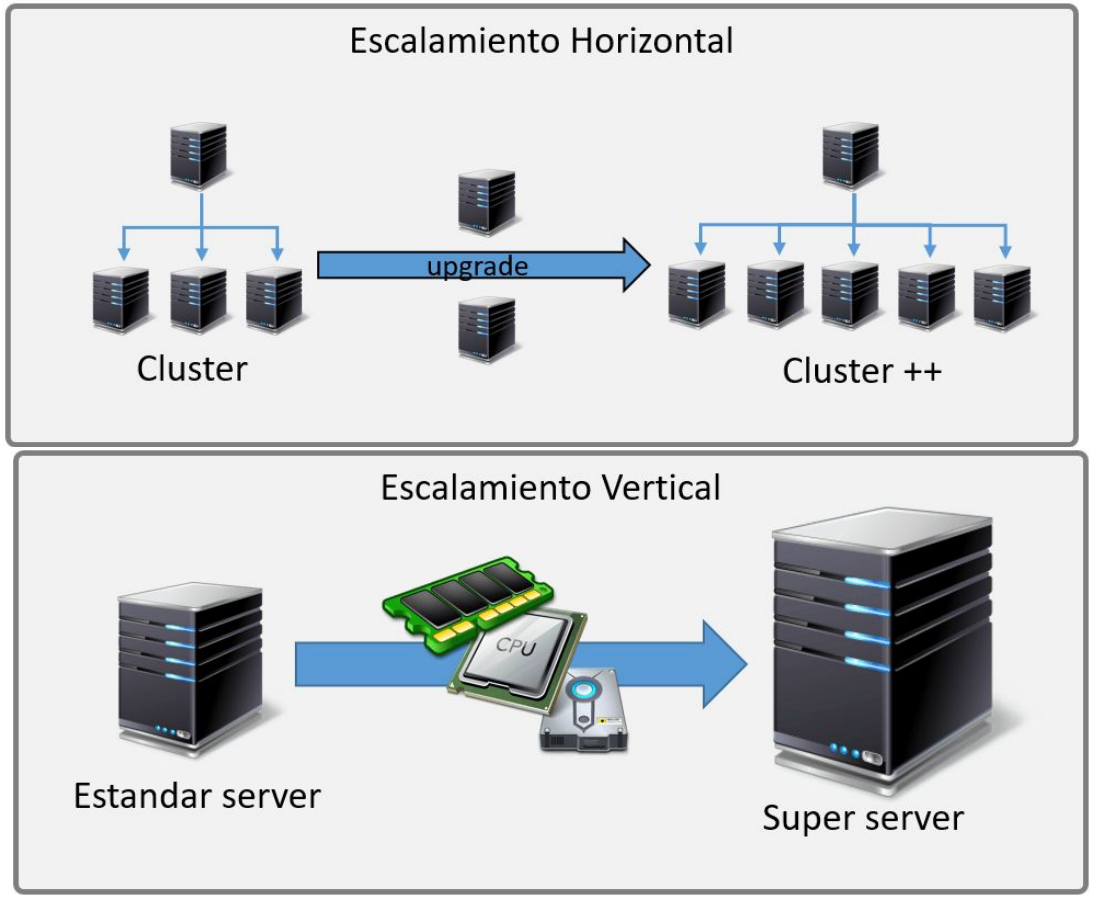
Los clústeres de computadoras son la base de la mayoría de las arquitecturas de procesamiento masivo en Big Data (como Hadoop y Apache Spark). **La escalabilidad se refiere a la capacidad del sistema para aumentar (o reducir) su capacidad de procesamiento y almacenamiento agregando (o eliminando) nodos en el clúster sin afectar el rendimiento general del sistema.**

Tipos de escalabilidad en clústeres:

- **Escalabilidad Horizontal:** Añadir más nodos (servidores) al clúster para incrementar su capacidad de procesamiento y almacenamiento. Si un clúster de Hadoop con 10 nodos está alcanzando su límite de rendimiento, se pueden añadir 5 nodos adicionales para aumentar la capacidad de procesamiento.
- **Escalabilidad Vertical:** Aumentar los recursos (CPU, RAM, almacenamiento) de los nodos existentes para mejorar el rendimiento. En lugar de añadir más nodos, se puede duplicar la cantidad de RAM en cada nodo para mejorar el procesamiento de datos intensivos en memoria.

Cluster de computadoras/ordenadores

Tipos de escalabilidad en clústeres:



Cluster de computadoras. Tolerancia a Fallos y Escalabilidad en Sistemas Distribuidos

Tolerancia a Fallos en HDFS

HDFS (Hadoop Distributed File System) es un ejemplo excelente de cómo los sistemas distribuidos implementan la **tolerancia a fallos**. H**DFS utiliza la replicación de datos para asegurar que la información permanezca accesible incluso si algunos nodos del sistema fallan.**

Características clave de tolerancia a fallos en HDFS:

1. **Replicación de bloques**: Por defecto, HDFS replica cada bloque de datos 3 veces.
2. **Distribución inteligente**: Las réplicas se almacenan en diferentes nodos y racks.
3. **Heartbeat y monitoreo**: El NameNode monitorea constantemente el estado de los DataNodes.
4. **Re-replicación automática**: Si se detecta la pérdida de una réplica, HDFS automáticamente crea una nueva.

Cluster de computadoras. Tolerancia a Fallos y Escalabilidad en Sistemas Distribuidos

Escalabilidad Horizontal en Sistemas Distribuidos

La escalabilidad horizontal se refiere a la capacidad de aumentar el rendimiento del sistema añadiendo más nodos al clúster, en lugar de aumentar la potencia de los nodos existentes (escalabilidad vertical).

Características de la escalabilidad horizontal:

1. **Adición fácil de nodos**: Nuevos servidores se pueden añadir al clúster según sea necesario.
2. **Distribución de carga**: El trabajo se distribuye entre todos los nodos disponibles.
3. **Rebalanceo de datos**: Los datos se redistribuyen para mantener una carga equilibrada.
4. **Aumento lineal de capacidad**: Cada nuevo nodo aumenta proporcionalmente la capacidad total.

Cluster de computadoras

Componentes clave para la escalabilidad:

- **Gestión de recursos**: Los clústeres utilizan gestores de recursos como **YARN** (Yet Another Resource Negotiator) o **Kubernetes** para distribuir las tareas entre los nodos de manera eficiente.
    - **YARN en Hadoop**: Actúa como el sistema operativo del clúster, gestionando los recursos de CPU y memoria de cada nodo.
    - **Kubernetes**: Coordina y escala aplicaciones distribuidas, manteniendo el equilibrio de carga y garantizando que los contenedores estén disponibles.
- **Equilibrio de carga**: La escalabilidad se mantiene a través del equilibrio de carga, distribuyendo las tareas de manera uniforme entre los nodos disponibles.
    - **Ejemplo**: Si un nodo en un clúster de Spark se sobrecarga, el gestor de recursos puede redirigir algunas de las tareas a otros nodos con menos carga.

Cluster de computadoras

Escalabilidad en acción: Apache Hadoop y Spark

- **Apache Hadoop**: Implementa la escalabilidad horizontal mediante el **Hadoop Distributed File System (HDFS)**. Los datos se dividen en bloques y se distribuyen a través de varios nodos. Si se añaden nuevos nodos, Hadoop reorganiza los bloques para aprovechar la nueva capacidad de almacenamiento y procesamiento.
- **Apache Spark**: **Utiliza un modelo de procesamiento en memoria** que permite realizar tareas a gran velocidad en clústeres escalables. La escalabilidad se logra agregando más nodos para procesar datos en paralelo.

Cluster de computadoras

Beneficios de la escalabilidad en clústeres:

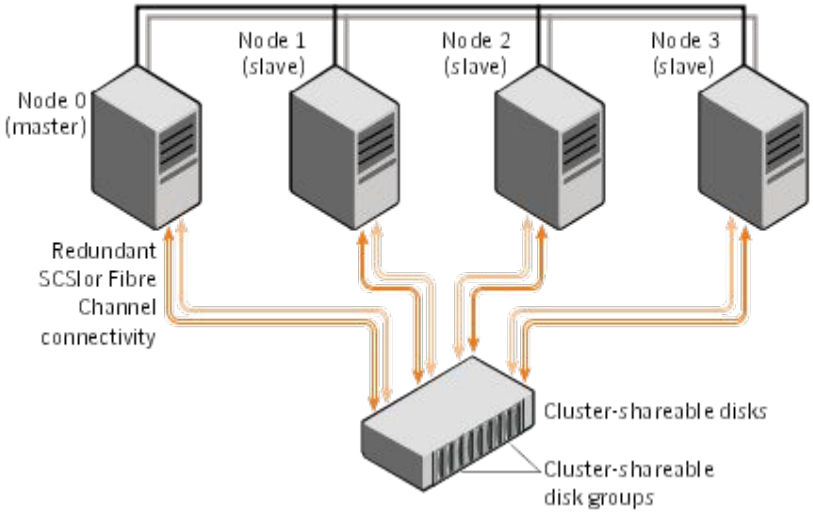
- **Reducción de tiempo de procesamiento**: La capacidad de distribuir las tareas entre múltiples nodos permite procesar grandes volúmenes de datos en paralelo, reduciendo significativamente el tiempo necesario para completar las tareas.
- **Ahorro de costos**: La escalabilidad horizontal permite a las empresas escalar sus recursos según la demanda, evitando inversiones costosas en hardware.
- **Flexibilidad**: Los clústeres escalables pueden adaptarse fácilmente a nuevas cargas de trabajo y cambiar en función de los requisitos del proyecto.

Cluster de computadoras

Ventajas:

Alto rendimiento: Los clusters **permiten acelerar cargas de trabajo al dividirlas en subtareas y distribuirlas entre los nodos,** lo que posibilita resolver problemas complejos de manera eficiente.

Alta disponibilidad: La monitorización constante de los nodos en el cluster permite detectar fallos y tomar medidas para mantener los servicios y datos disponibles, ya sea reiniciando nodos caídos o respondiendo desde réplicas.

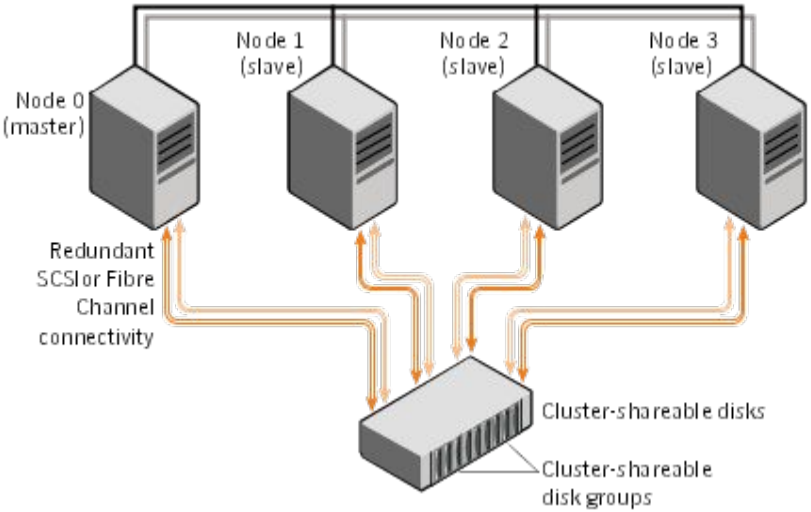


Cluster de computadoras

Ventajas:

Equilibrio de carga: Los algoritmos distribuyen las cargas de trabajo entre los nodos para evitar cuellos de botella, considerando factores como el tamaño del trabajo y la capacidad de procesamiento de cada nodo.

Escalabilidad: La **capacidad de añadir nuevos nodos al cluster permite aumentar la potencia de cálculo de manera flexible,** sin necesidad de estimaciones previas.



Almacenamiento. Tipos

Bases de Datos Relacionales:

Son **sistemas de almacenamiento de datos estructurados que utilizan tablas para organizar la información.** Cada tabla contiene filas y columnas, donde **cada fila representa un registro único y cada columna representa un atributo específico.** Estas bases de datos son conocidas por su estructura tabular y la capacidad de establecer relaciones entre diferentes tablas a través de claves primarias y claves foráneas.

Ejemplos populares de sistemas de gestión de bases de datos relacionales (RDBMS) incluyen MySQL, PostgreSQL y Microsoft SQL Server.

- Gestión de Empleados: Las empresas utiliza una base de datos relacionales para almacenar información sobre sus empleados (nombres, apellidos, edades y departamentos).

Almacenamiento. Tipos

Datasets:

Un dataset es un c**onjunto de datos que se agrupa en una colección organizada. Puede contener datos de cualquier tipo, como texto, números, imágenes o información en formato tabular.** Los datasets se utilizan en diversas aplicaciones, desde **análisis de datos hasta aprendizaje automático.** Los datos dentro de un dataset suelen tener un propósito específico, como la investigación científica o la toma de decisiones empresariales.

- Análisis de Ventas Mensuales: Un analista utiliza un dataset que contiene datos mensuales de ventas para realizar un análisis de tendencias y tomar decisiones estratégicas sobre estrategias de marketing y gestión de inventario.

Almacenamiento. Tipos

Almacenes de Datos (Data Warehouses):

Infraestructura de almacenamiento diseñada para recopilar, almacenar y gestionar grandes volúmenes de datos de diferentes fuentes. Su objetivo principal es proporcionar una única fuente de verdad para el análisis de datos empresariales. Los almacenes de datos suelen estar optimizados para consultas y análisis, lo que facilita la obtención de información valiosa de los datos almacenados.

Análisis Empresarial: Una empresa utiliza un almacén de datos para consolidar información de ventas, inventario y finanzas de todas sus sucursales. Esto permite a los directivos acceder a informes detallados para la toma de decisiones empresariales.

Almacenamiento. Propiedades

ACID vs. CAP vs. BASE:

ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad): Es un **conjunto de propiedades que garantizan la integridad de las transacciones en una base de datos relacional.**

- **Atómicas:** se ejecutan en su totalidad o no se ejecutan en absoluto.
- **Consistentes:** mantienen la integridad de la base de datos.
- **Aisladas:** las transacciones no interfieren entre sí.
- **Duraderas:** los cambios persisten incluso en caso de fallo del sistema.

CAP (Consistencia, Disponibilidad, Tolerancia a partición): El teorema CAP establece que en un sistema distribuido, es imposible garantizar simultáneamente la consistencia, la disponibilidad y la tolerancia a partición en caso de una falla de red.

Almacenamiento. Propiedades

ACID vs. CAP vs. BASE:

BASE (Básicamente Disponible, Suave, Eventualmente consistente): En contraposición a ACID, BASE se refiere a sistemas que priorizan la disponibilidad y la tolerancia a partición en sistemas distribuidos. Los sistemas BASE pueden ser "eventualmente consistentes", lo que significa que, con el tiempo, todos los nodos en el sistema llegarán a un estado consistente.

Analogías:

- ACID es como un cajero automático: cada transacción es completa (atómica), mantiene el balance correcto (consistente), las transacciones no se mezclan (aisladas), y tu dinero está seguro incluso si hay un corte de luz (durable).
- CAP es como elegir entre tres sabores de helado, pero sólo puedes tener dos: consistencia (vainilla), disponibilidad (chocolate) y tolerancia a partición (fresa).
- BASE es como un grupo de WhatsApp: los mensajes llegan a todos eventualmente, pero no siempre al mismo tiempo o en el mismo orden.

Almacenamiento. Tipos

Bases de Datos NoSQL y Orientadas a Grafos:

Bases de Datos NoSQL: Estas bases de datos **se utilizan para gestionar datos no estructurados o semiestructurados, como documentos, gráficos o datos de series temporales.** No siguen el modelo relacional tradicional y permiten una mayor escalabilidad y flexibilidad en la gestión de datos. Ejemplos incluyen MongoDB (base de datos de documentos) y Cassandra (base de datos de columnas ampliamente distribuida).

Bases de Datos Orientadas a Grafos: Estas bases de datos **se utilizan para modelar y gestionar datos con relaciones complejas, como redes sociales o sistemas de recomendación.** Utilizan estructuras de grafo para representar y almacenar datos, lo que facilita la búsqueda y navegación eficiente de relaciones. Ejemplos incluyen Neo4j y Amazon Neptune.

Almacenamiento. Tipos. Ventajas y desventajas

Aspecto	Bases de Datos Relacionales	Bases de Datos NoSQL
Estructura	Esquema fijo y predefinido	Esquema flexible y dinámico
Tipo de datos	Datos estructurados	Datos estructurados, semiestructurados y no estructurados
Escalabilidad	Escalabilidad vertical (añadir más recursos a un servidor)	Escalabilidad horizontal (añadir más nodos)
Consistencia	Alta consistencia	Consistencia eventual en muchos casos
Rendimiento	Optimizadas para transacciones	Alta eficiencia en lectura/escritura
Casos de uso	Aplicaciones financieras, CRM	Aplicaciones web, redes sociales, IoT

Almacenamiento. Tipos. Ventajas y desventajas

Análisis Comparativo: MongoDB vs. Cassandra en Arquitecturas Distribuidas

MongoDB

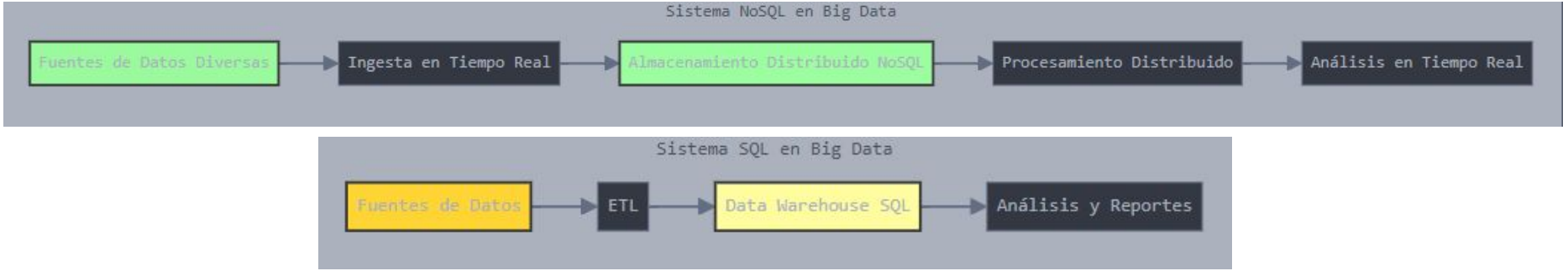
- **Modelo de datos:** Orientado a documentos
- **Consistencia:** Eventual con opciones de consistencia fuerte
- **Escalabilidad:** Horizontal mediante sharding
- **Consultas:** Lenguaje de consulta flexible y potente
- **Casos de uso:** Aplicaciones web, móviles, análisis en tiempo real

Cassandra

- **Modelo de datos:** Orientado a columnas
- **Consistencia:** Ajustable (desde eventual hasta fuerte)
- **Escalabilidad**: Lineal y horizontal
- **Consultas:** CQL (similar a SQL, pero con limitaciones)
- **Casos de uso:** IoT, sistemas de mensajería, aplicaciones de alta disponibilidad

Almacenamiento. Tipos. Ventajas y desventajas

Análisis Comparativo: MongoDB vs. Cassandra en Arquitecturas Distribuidas. Aplicabilidad en Arquitecturas Distribuidas



Almacenamiento. Tipos. Ventajas y desventajas

Almacenamiento en bloques:

- Almacena los datos en bloques individuales, similar a los discos duros tradicionales. Ideal para aplicaciones que requieren acceso rápido y aleatorio a grandes volúmenes de datos.
- **Ejemplo**: **Amazon EBS** (Elastic Block Storage), **Google Persistent Disk**.

Almacenamiento de objetos:

- Almacena los datos como objetos individuales con metadatos, ideal para grandes volúmenes de datos no estructurados. Se utiliza comúnmente en **Data Lakes** y para almacenar imágenes, videos, documentos, y logs.
- **Ejemplo**: **Amazon S3**, **Azure Blob Storage**, **Google Cloud Storage**.

Almacenamiento en archivos:

- Los datos se almacenan y organizan en estructuras de archivos (carpetas y subcarpetas). Ideal para aplicaciones que requieren sistemas de archivos compartidos.
- **Ejemplo**: **Amazon EFS** (Elastic File System), **Google Cloud Filestore**.

Almacenamiento. Almacenamiento en la nube

Ventajas del almacenamiento en la nube para Big Data:

- **Escalabilidad dinámica**: Aumenta o reduce la capacidad de almacenamiento según las necesidades.
- **Bajo coste de mantenimiento**: No se requieren inversiones en infraestructura física.
- **Disponibilidad global**: Permite el acceso a los datos desde cualquier lugar, facilitando el trabajo remoto y la colaboración.
- **Integración con herramientas de análisis**: Los servicios en la nube como **AWS Athena** y **Google BigQuery** permiten realizar análisis de datos directamente sobre el almacenamiento en la nube sin necesidad de mover los datos.

Almacenamiento. Almacenamiento en la nube

Almacenamiento en la Nube y NoSQL

Las empresas modernas están aprovechando la combinación de almacenamiento en la nube y bases de datos NoSQL para manejar grandes volúmenes de datos no estructurados y realizar análisis en tiempo real.

Uso de S3 y Azure Blob Storage

1. **Amazon S3 (Simple Storage Service)**:
    - Almacenamiento de objetos altamente escalable
    - Ideal para datos no estructurados como logs, imágenes, videos
    - Integración nativa con servicios de AWS como Athena para consultas SQL ad-hoc
2. **Azure Blob Storage**:
    - Servicio de almacenamiento de objetos de Microsoft
    - Optimizado para almacenar grandes cantidades de datos no estructurados
    - Se integra con Azure Data Lake Analytics para procesamiento y análisis

Almacenamiento. Procesamiento y recuperación de datos

Las bases de datos tradicionales están basadas generalmente en sistemas relacionales u objeto-relacionales. **Para el acceso, procesamiento y recuperación de los datos, se sigue el modelo Online Transaction Processing (OLTP).** El modelo OLTP (procesamiento de transacciones en línea), permite gestionar los cambios de la base de datos mediante la **inserción, actualización y eliminación** de información de la misma a través de transacciones básicas que son procesadas en tiempos muy pequeños.

Con respecto a la recuperación de información de la base de datos, se utilizan operadores clásicos (concatenación, proyección, selección, agrupamiento...) para realizar consultas básicas y sencillas (realizadas, mayoritariamente, en lenguaje SQL y extensiones del mismo).

Almacenes de datos. Bases de datos tradicionales.

Las **bases de datos relacionales** son colecciones de datos integrados, almacenados en un soporte secundario no volátil y con redundancia controlada. La definición de los datos y la estructura de la base de datos debe estar basada en un **modelo de datos** que permita captar las interrelaciones y restricciones existentes en el dominio que se pretende modelizar.

A su vez, un Sistema Gestor de Bases de Datos se compone de una colección de datos estructurados e interrelacionados (una base de datos) así como de un conjunto de programas para acceder a dichos datos.

Almacenes de datos. Bases de datos tradicionales.

La revolución en la generación, almacenamiento y procesamiento de los datos, así como **la irrupción del big data, han puesto a prueba el modelo de funcionamiento, rendimiento y escalabilidad de las bases de datos relacionales tradicionales.**

En este sentido, la inteligencia de negocio, más conocida por el término inglés **business intelligence,** investiga en el diseño y desarrollo de este tipo de soluciones. La **inteligencia de negocio puede definirse como la capacidad de una empresa de estudiar sus acciones y comportamientos pasados para entender dónde ha estado la empresa, determinar la situación actual y predecir o cambiar lo que sucederá en el futuro,** utilizando las soluciones tecnológicas más apropiadas para optimizar el proceso de toma de decisiones.

Almacenes de datos. Bases de datos tradicionales.

Actividad: Business Intelligence.

- Lee el siguiente artículo.

https://cloud.google.com/learn/what-is-business-intelligence?hl=es#:~:text=La%20inteligencia%20empresarial%20(BI)%20es,de%20decisiones%20estrat%C3%A9gicas%20y%20cotidianas.

- Reflexiona sobre el contenido para participar en el debate grupal.

Almacenes de datos. Bases de datos tradicionales.

Actividad: Business Intelligence.

Existen tres categorías principales de herramientas de inteligencia empresarial:

- **On-premise:** Instaladas en la infraestructura interna de la organización. Son menos escalables que las soluciones basadas en la nube.
- **De código abierto:** Económicas, pero requieren conocimientos técnicos avanzados y programación manual.
- **Basadas en la nube:** Especialmente útiles para gestionar grandes volúmenes de datos y datos en tiempo real. Ofrecen rentabilidad al tercerizar la infraestructura y el soporte técnico.

Ventajas de la inteligencia empresarial:

- **Mejora del rendimiento empresarial:** Permite a las empresas interpretar qué ha sucedido, por qué, y cómo optimizar las operaciones.
- **Toma de decisiones basadas en indicadores clave de rendimiento (KPI):** Ayuda a extraer conclusiones y tomar medidas prácticas para influir en el rendimiento.
- **Identificación de oportunidades:** Facilita la adaptación a cambios en el mercado y la identificación de nuevas líneas de negocio o clientes.

Importancia de la inteligencia empresarial:

Es clave para optimizar la toma de decisiones y el rendimiento, al identificar patrones y tendencias en los datos empresariales. Sin embargo, su efectividad depende de la calidad de los datos, la integración entre sistemas y los conocimientos técnicos necesarios para usar las herramientas BI.

Almacenes de datos. Bases de datos tradicionales.

Actividad: Business Intelligece.

Usos de la inteligencia empresarial:

- Informes: Datos resumidos para la toma de decisiones estratégicas.
- Visualización de datos: Facilita la comprensión de datos complejos.
- Analíticas predictivas: Predicción de patrones futuros basados en datos históricos.
- Minería de datos: Identificación de tendencias y patrones útiles.
- Procesamiento de eventos complejos: Análisis de datos en tiempo real.
- Gestión del rendimiento empresarial: Medición de objetivos operativos.

Almacenes de datos. Modelos de procesamiento.

Procesamiento en paralelo y distribuido

El procesamiento en paralelo aprovecha las capacidades de los procesadores multinúcleo actuales para ejecutar diferentes hilos de forma concurrente. El sistema operativo reparte el tiempo de CPU entre los hilos.

El procesamiento distribuido utiliza un clúster de máquinas conectadas en red. Divide la carga de trabajo en subtareas para ejecutar en paralelo en los distintos nodos. Requiere comunicación de datos entre nodos, siendo más rápida dentro del mismo equipo que entre equipos.

Almacenes de datos. Modelos de procesamiento.

Estrategias de procesamiento

- **Batch:** Procesamiento sin restricciones de tiempo, típicamente para analítica. Puede tardar horas o días. Se aplica sobre toda la cantidad de datos.
- **Transaccional:** Requiere tiempos de respuesta cortos, por debajo del segundo. Se usa en operaciones de OLTP con bases de datos relacionales.
- **Tiempo real:** Procesamiento de baja latencia para analítica interactiva con usuarios. Se suele usar en sistemas OLAP.
- **Streaming**: Debe procesar datos entrantes a la velocidad que llegan. Obliga a estructuras de datos actualizables en memoria.
- **OLTP**: Orientado a procesamiento de transacciones en línea. Usa bases de datos relacionales y tiempos de respuesta cortos.
- **OLAP:** Orientado a consultas analíticas en tiempo real. Usa bases de datos multidimensionales optimizadas para consultas complejas.

Almacenes de datos. Modelos de procesamiento.

Principio SCV

Establece que un sistema de procesamiento distribuido sólo puede tener 2 de estas 3 propiedades:

- Velocidad: tiempo de procesamiento desde que se reciben los datos.
- Consistencia: la precisión y coherencia de los resultados, lo cual puede depender de si se utilizan todos los datos disponibles o solo muestras.
- Volumen: cantidad de datos que se pueden procesar. en un tiempo determinado.

Un sistema distribuido puede ser rápido y preciso pero no manejar grandes volúmenes de datos, o puede ser preciso y manejar grandes volúmenes de datos pero no ser muy rápido. La elección de estas propiedades depende de las necesidades y las limitaciones del sistema.

Almacenes de datos. Modelos de procesamiento.

Los principales elementos involucrados en el procesamiento de datos en entornos de Big Data son:

- Fuentes de datos: bases de datos transaccionales, archivos, sensores, aplicaciones, redes sociales, etc.
- Ingestión y recolección de datos: procesos para conectarse a las fuentes, extraer los datos y unificarlos.
- Almacenamiento: sistemas como HDFS, lago de datos (DataLake), bases NoSQL, etc. .
- Procesamiento distribuido: plataformas como Hadoop, Spark, Flink, etc.
- Estrategias de procesamiento: batch, transaccional, tiempo real, streaming.

Almacenes de datos. Modelos de procesamiento.

Los principales elementos involucrados en el procesamiento de datos en entornos de Big Data son:

- Modelos de programación: MapReduce, SQL, Python, R. Formas de expresar el procesamiento a realizar.
- Infraestructura: clústers de nodos, redes de interconexión, sistemas de archivos distribuidos.
- Seguridad: control de acceso, encriptación, anonimización, etc.
- Gobernanza: definir políticas, calidad de datos, linajes, diccionarios.
- Analítica y visualización: para extraer insights y presentar resultados.

DATA SOURCES

Internal data sources such as data from CRM system, ERP system, sales reports, etc.

External data sources such as government statistics and media channels

DATA STORAGE

Big data storage software tools store, manage and retrieve massive amounts of data.

APACHE hadoop

mongoDB

cassandra

DATA MINING

Data mining tools allow businesses to extract usable data from a huge set of raw data to find relationships, patterns, and anomalies.

rapidminer

SPSS Modeler

DATA ANALYTICS

Although data mining tools incorporate data analysis, there are software designed specifically with advanced analytical capabilities.

APACHE Spark

kafka

IBM Analytics

DATA VISUALIZATION

Data visualization software is also a type of data analytics tool. However, they are specifically designed to take the raw data and presenting it with beautiful and easy digestible visuals like graphs and charts.

+ a b | e a u

looker

Klipfolio

Almacenes de datos. Modelos de procesamiento.

Estas nuevas soluciones requerirán un modelo de procesamiento diferente a OLTP. Esto es así, ya que el objetivo perseguido por la inteligencia de negocio está menos orientado al ámbito transaccional y más enfocado al ámbito analítico.

Las nuevas soluciones utilizan el modelo Online analytical processing (OLAP). La principal diferencia entre OLTP y OLAP estriba en que mientras que el primero es un sistema de procesamiento de transacciones en línea, el segundo es un sistema de recuperación y análisis de datos en línea.

Por tanto, OLAP complementa a SQL aportando la capacidad de analizar datos desde distintas variables y dimensiones, mejorando el proceso de toma de decisiones.

Almacenes de datos. Modelos de procesamiento. OLAP

Los sistemas OLAP están basados, generalmente, en sistemas o interfaces multidimensionales que proporcionan facilidades para la transformación de los datos, permitiendo obtener nuevos datos más combinados y agregados que los obtenidos mediante las consultas simples realizadas por OLTP. Al contrario que en OLTP, las unidades de trabajo de OLAP son más complejas que en OLTP y consumen más tiempo.

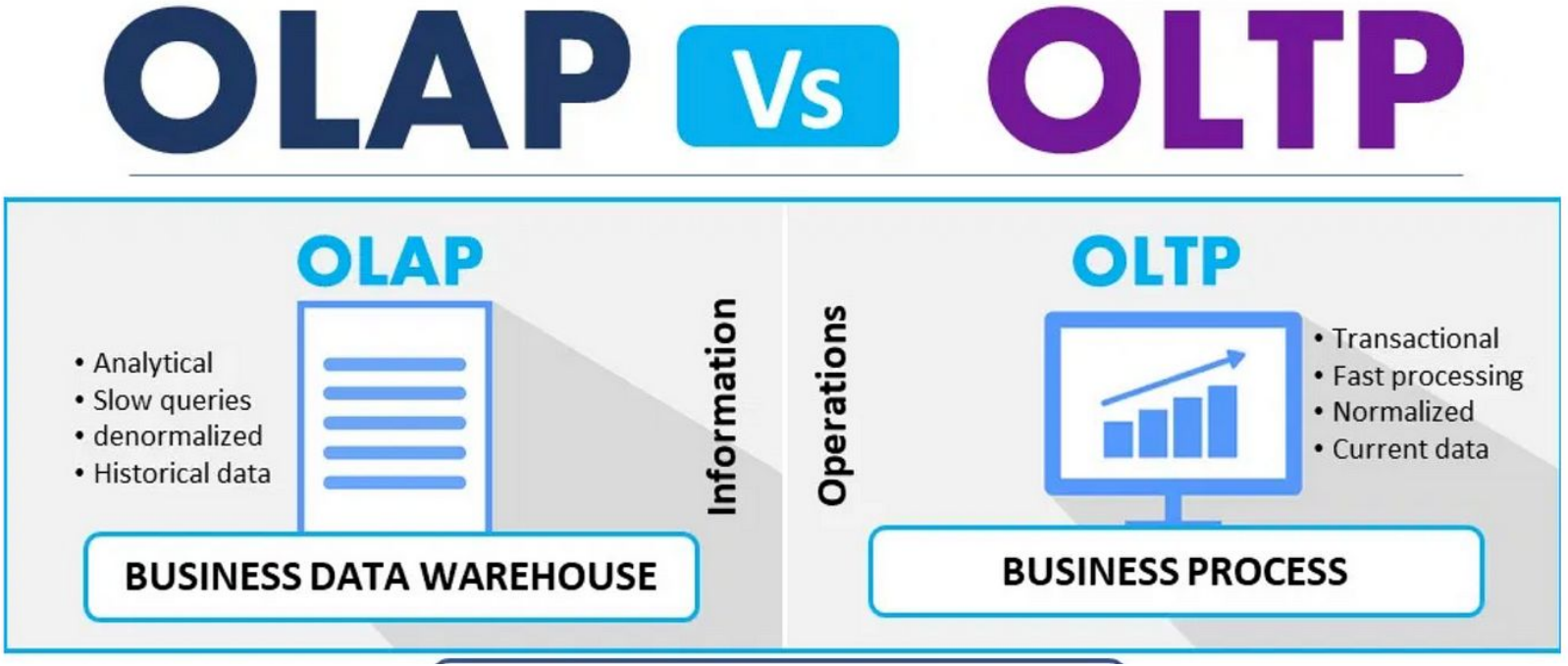
En cuanto a la visualización de los mismos, los sistemas OLAP permiten la visualización y el análisis multidimensional a partir de diferentes vistas de los datos, presentando los resultados en forma matricial y con mayores posibilidades estéticas y visuales.

Almacenes de datos. Modelos de procesamiento. OLAP vs OLTP

Tabla 1: Tabla resumen y comparativa entre OLTP y OLAP

	Bases de datos relacionales	Soluciones Business Intelligence
	OLTP	**OLAP**
Concepto	Sistema de procesamiento de transacciones en línea	Sistema de recuperación y análisis de datos en línea
Funciones	Gestión de transacciones: inserción, actualización, eliminación...	Análisis de datos para dar soporte a la toma de decisiones
Procesamiento	Transacciones cortas	Procesamientos de análisis complejos
Tiempo	Las transacciones requieren poco tiempo de ejecución	Los análisis requieren mayor tiempo de ejecución
Consultas	Simples, utilizando operadores básicos tradicionales	Complejas, permitiendo analizar los datos desde múltiples dimensiones
Visualización	Básica. Muestra los datos en forma tabular	Muestra los datos en forma matricial. Mayores posibilidades gráficas

Almacenes de datos. Modelos de procesamiento. OLAP vs OLTP



Almacenes de datos. Modelos de procesamiento. OLAP vs OLTP

OLTP (Procesamiento de Transacciones en Línea):

Propósito: OLTP se utiliza para procesar transacciones comerciales en tiempo real. Su objetivo principal es mantener la integridad y consistencia de los datos transaccionales.

Ejemplos:

- **Pedidos:** Cuando un cliente realiza una compra en línea, el sistema OLTP registra la transacción, actualiza el inventario y procesa el pago.
- **Reservas:** En una aerolínea, el sistema OLTP maneja las reservas de vuelos, asientos y emite boletos.
- **Actualizaciones de cuentas:** Los bancos utilizan OLTP para registrar depósitos, retiros y transferencias entre cuentas.

OLAP (Procesamiento Analítico en Línea):

Propósito: OLAP se centra en el análisis de datos agregados desde diferentes perspectivas. Ayuda a tomar decisiones estratégicas basadas en información procesable.

Ejemplos:

- **Informes financieros:** Un equipo financiero utiliza OLAP para analizar los ingresos, gastos y tendencias financieras de la empresa.
- **Minería de datos:** Los científicos de datos utilizan OLAP para descubrir patrones ocultos en grandes conjuntos de datos.
- **Planificación estratégica:** Una cadena minorista utiliza OLAP para evaluar el rendimiento de las tiendas, identificar productos populares y optimizar la colocación.

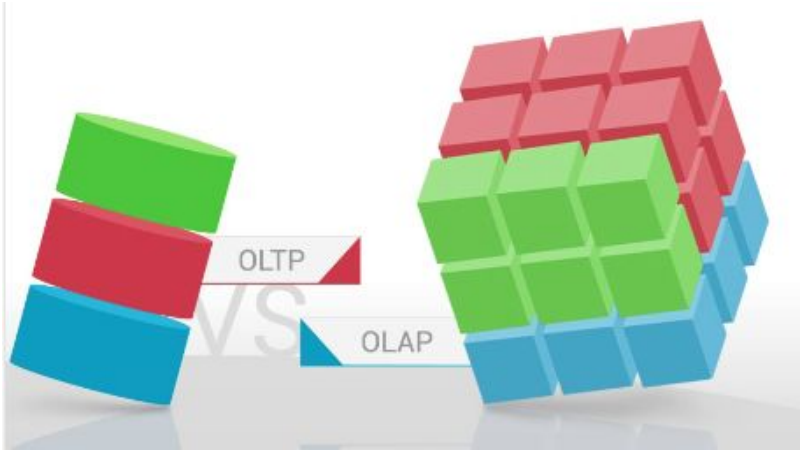
Almacenes de datos. Modelos de procesamiento. OLAP vs OLTP

OLTP se enfoca en transacciones individuales y actualizaciones constantes de datos, mientras que **OLAP se centra en análisis complejos y toma de decisiones estratégicas**. Ambos sistemas son esenciales para el funcionamiento eficiente de una organización.

Almacenes de datos. Modelos de procesamiento. OLAP vs OLTP

La arquitectura del software OLAP incluye dos componentes básicos:

Servidor OLAP: proporciona almacenamiento de datos, realizando sobre ellos las operaciones necesarias y la formación de un modelo multidimensional a nivel conceptual.
Cliente de procesamiento analítico en línea: presenta al usuario una interfaz para el modelo de datos multidimensional, brindándole la capacidad de manipular datos convenientemente para realizar tareas de análisis.



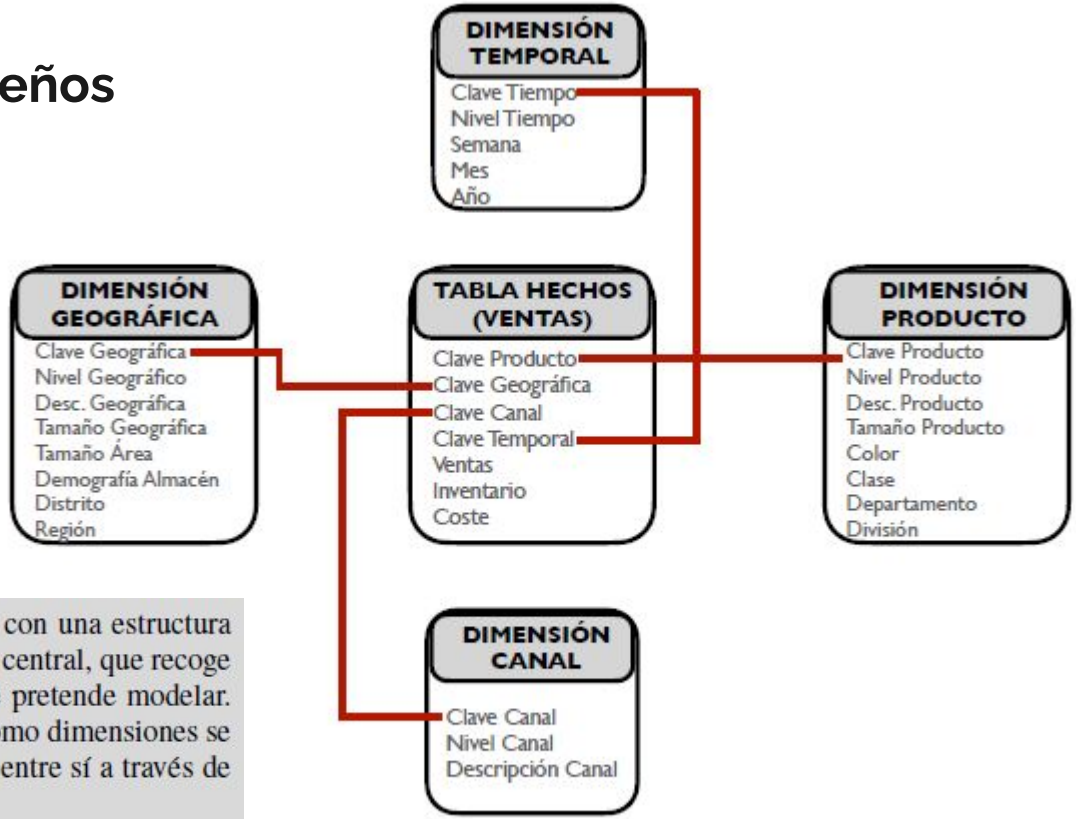
Almacenes de datos. Diseños

A la hora de diseñar un almacén de datos, existen dos alternativas ampliamente utilizadas: el diseño en estrella, que promueve el diseño directo de estructuras lógicas sobre el modelo relacional, y el diseño en copo de nieve, como variante del diseño en estrella.

Star Schema

Snowflake Schema

Almacenes de datos. Diseños



El diseño en estrella permite crear un almacén de datos con una estructura centralizada. El diseño se compone de una tabla de hechos central, que recoge los valores de las medidas del proceso de negocio que se pretende modelar. Rodeando a la tabla de hechos, se incluyen tantas tablas como dimensiones se hayan especificado en el modelo, las cuales se relacionan entre sí a través de la tabla de hechos.

Figura 2.5: Ejemplo de almacén de datos diseñado en estrella.

Almacenes de datos. Diseños

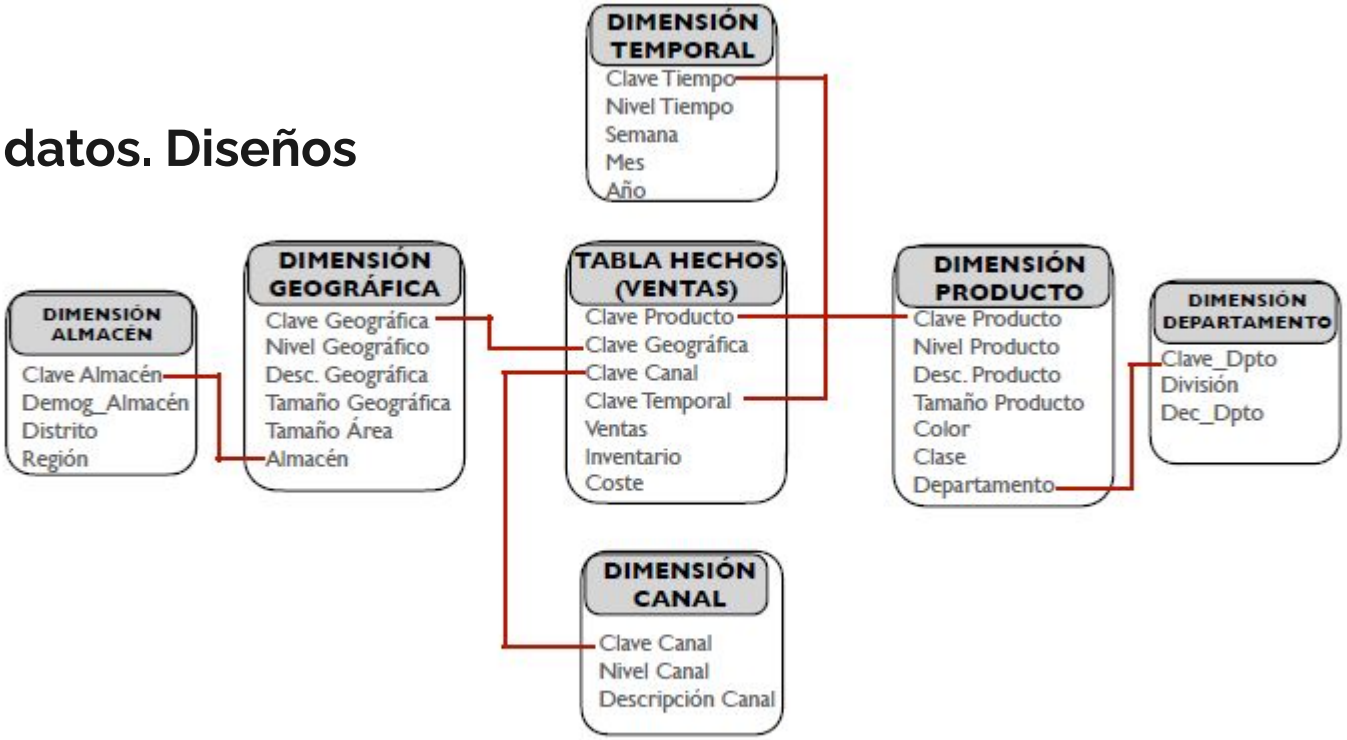
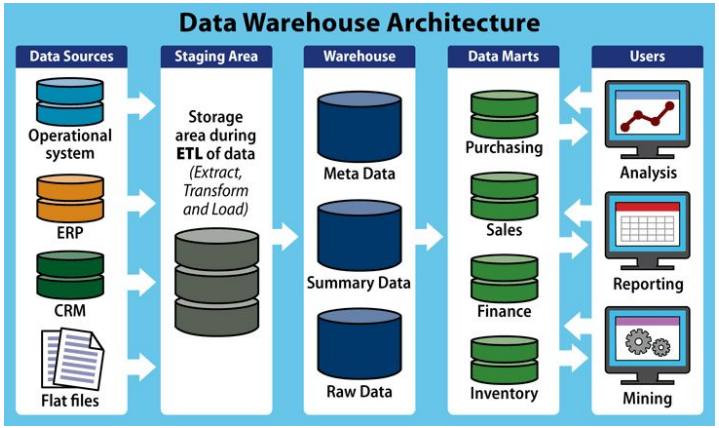


Figura 2.6: Ejemplo de almacén de datos diseñado en copo de nieve.

El diseño en copo de nieve permite crear un almacén de datos a partir de un diseño en estrella. Para ello, las tablas de dimensiones se normalizan, generando más de una tabla para cada una de las dimensiones, mejorando la eficiencia de consultas complejas que requieren el uso de operadores avanzados.

Almacenes de datos. Datawarehouse

Un almacén de datos, más conocido por el término data warehouse (en inglés), es una solución de business intelligence que combina tecnologías y componentes con el objetivo de ayudar al uso estratégico de los datos por parte de una organización. Esta solución debe proveer a la empresa, de forma integrada, de capacidad de almacenamiento de una gran cantidad de datos así como de herramientas de análisis de los mismos que, frente al procesamiento de transacciones, permita transformar los datos en información para ponerla a disposición de la organización y optimizar el proceso de toma de decisiones.

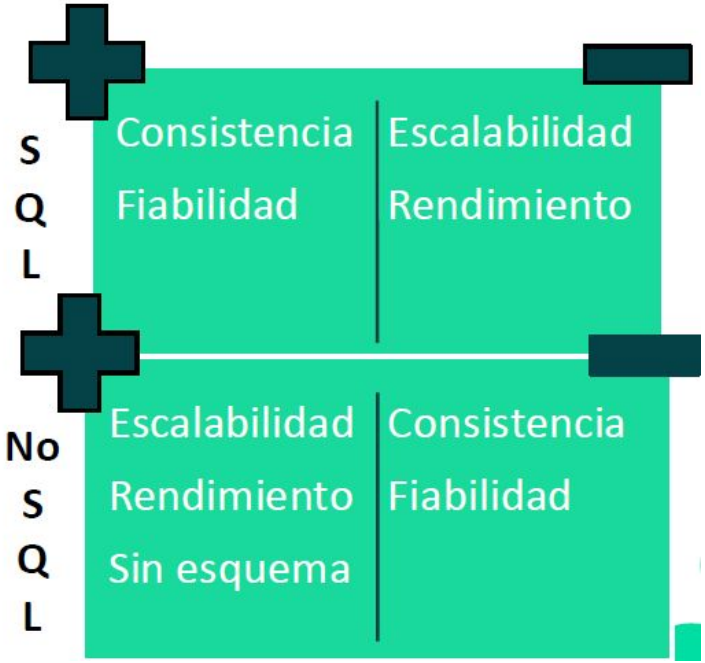


Almacenes de datos. Bases de datos documentales u orientadas a documentos.

La gran variedad y heterogeneidad en los tipos de datos almacenados y procesados en los últimos años ha puesto puesto en cuestión de si las bases de datos relacionales son el modelo más óptimo para trabajar con según qué tipos de datos. Como alternativa a ellas, en los últimos años han proliferado las bases de datos NoSQL.

NoSQL es el término utilizado para referirse a un tipo de bases de datos que permiten almacenar y gestionar tipos de datos que tradicionalmente han sido difíciles de gestionar por parte de las bases de datos relacionales. Así pues, NoSQL hace referencia a bases de datos documentales, bases de datos orientadas a grafos, buscadores, etc.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.



Almacenes de datos. Bases de datos documentales u orientadas a documentos.

Bases de datos SQL	Bases de datos NoSQL
Son bases de datos relacionales	Son bases de datos no relacionales o distribuidas
Utilizan lenguaje de consulta estructurado (SQL) y tienen un esquema predefinido	Tienen esquemas dinámicos para datos no estructurados
Son mejores para transacciones con múltiples filas	Son mejores para datos no estructurados como documentos o JSON
Son eficientes, flexibles y pueden ser accedidas fácilmente por cualquier aplicación	Ofrecen escalabilidad horizontal, lo que significa que simplemente se deben agregar más servidores para aumentar su carga de datos
Tienen esquemas rígidos, complejos y tabulares y típicamente requieren una escalabilidad vertical costosa	Son más flexibles y pueden manejar una variedad de tipos de datos
Son una buena opción cuando se trabaja con datos relacionados	Son útiles para infraestructuras modernas basadas en la nube
Ejemplos: MySQL, Oracle, PostgreSQL	Ejemplos: MongoDB, Cassandra, Couchbase, Amazon DynamoDB, Redis

Almacenes orientadas a la estructura

Las arquitecturas orientadas a la estructura reciben su nombre debido a que están diseñadas poniendo especial énfasis en el número de capas y elementos que componen la arquitectura del sistema de almacén de datos.

- **Arquitectura de una capa:** Esta arquitectura busca minimizar el almacenamiento de datos al eliminar redundancias. Su simplicidad impide cumplir con la propiedad de separación, ya que los análisis se realizan directamente sobre los datos operativos.

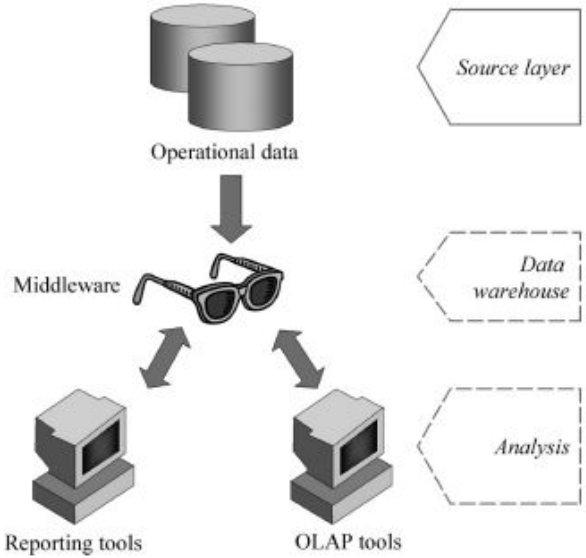


Figura 2.1: Almacén de datos. Arquitectura de una capa.

Almacenes orientadas a la estructura

Arquitectura de Una Capa: En una arquitectura de una sola capa, tanto los datos transaccionales como los datos analíticos se almacenan en una única base de datos sin separación entre ellos.

- Esta arquitectura es útil para aplicaciones pequeñas que no requieren análisis complejos y donde la cantidad de datos es relativamente baja.
- Una pequeña empresa minorista que desea generar informes de ventas diarios y semanales a partir de un conjunto de datos pequeño. En este caso, la arquitectura de una capa permite una implementación sencilla sin necesidad de crear estructuras adicionales.

Ventajas:

- Menor complejidad y bajo coste de implementación.
- Fácil acceso a los datos operativos en tiempo real.

Desventajas:

- Limitada escalabilidad y dificultades para separar los datos históricos de los operacionales.
- Poco eficiente para análisis de grandes volúmenes de datos.

Almacenes orientadas a la estructura

- **Arquitectura de dos capas:** Diseñada con el objetivo de solucionar el problema de la separación que presentaba la arquitectura de una capa. El esquema consigue subrayar la separación entre los datos disponibles y el almacén de datos a través de los siguientes componentes:
  - **Capa de origen (fuente)**
  - **Puesta a punto**
  - **Capa de almacén de datos**
  - **Análisis**

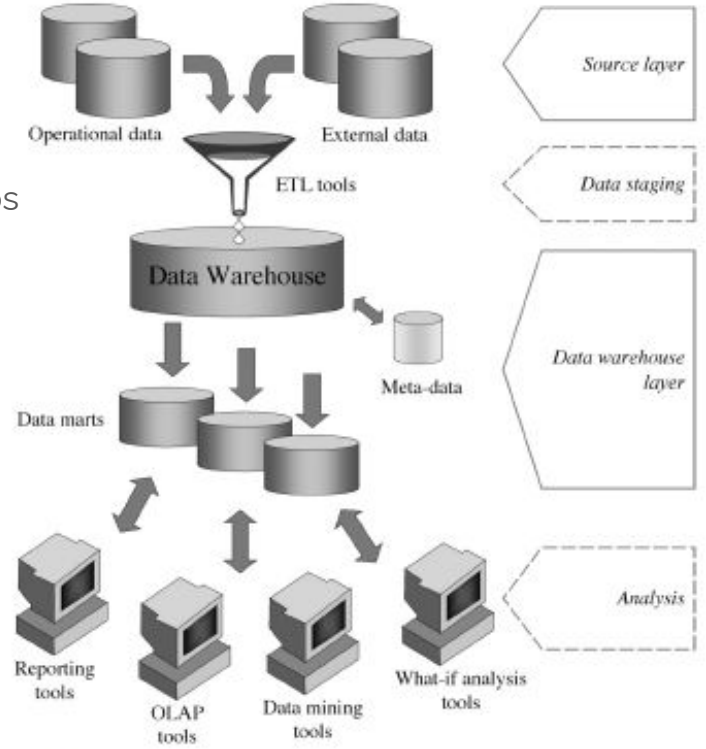


Figura 2.2: Almacén de datos. Arquitectura de dos capas.

Almacenes orientadas a la estructura

Arquitectura de Dos Capas: En una arquitectura de dos capas, **los datos se separan en una capa transaccional (OLTP) y una capa analítica o de almacén de datos (OLAP)**. Los datos se extraen, transforman y cargan (ETL) de la capa transaccional a la capa analítica.

- Ideal para organizaciones medianas que necesitan realizar análisis periódicos o informes sobre datos consolidados sin afectar las operaciones diarias.
- Ejemplo: Una empresa de marketing que recopila datos de campañas publicitarias en una base de datos OLTP y posteriormente transfiere estos datos a un almacén OLAP para realizar análisis de rendimiento de campañas, segmentación de clientes y proyecciones de ventas.

Ventajas:

- Mejora el rendimiento del análisis, ya que la capa analítica está optimizada para consultas complejas.
- Separación clara entre los datos operativos y los de análisis, lo que permite una mayor consistencia y seguridad en la capa de datos transaccionales.

Desventajas:

- Mayor complejidad que la arquitectura de una capa.
- Costos adicionales por el mantenimiento de dos capas separadas y procesos ETL.

Almacenes orientadas a la estructura

- **Arquitectura de tres capas:** Incluye una capa llamada de datos reconciliados o almacén de datos operativos. Con esta capa, **los datos operativos obtenidos tras la limpieza y depuración son integrados y validados, proporcionando un modelo de datos de referencia** para toda la organización.

Esta capa de datos reconciliados también puede implementarse de forma virtual en una arquitectura de dos capas, ya que se define como una vista integrada y coherente de los datos de origen.

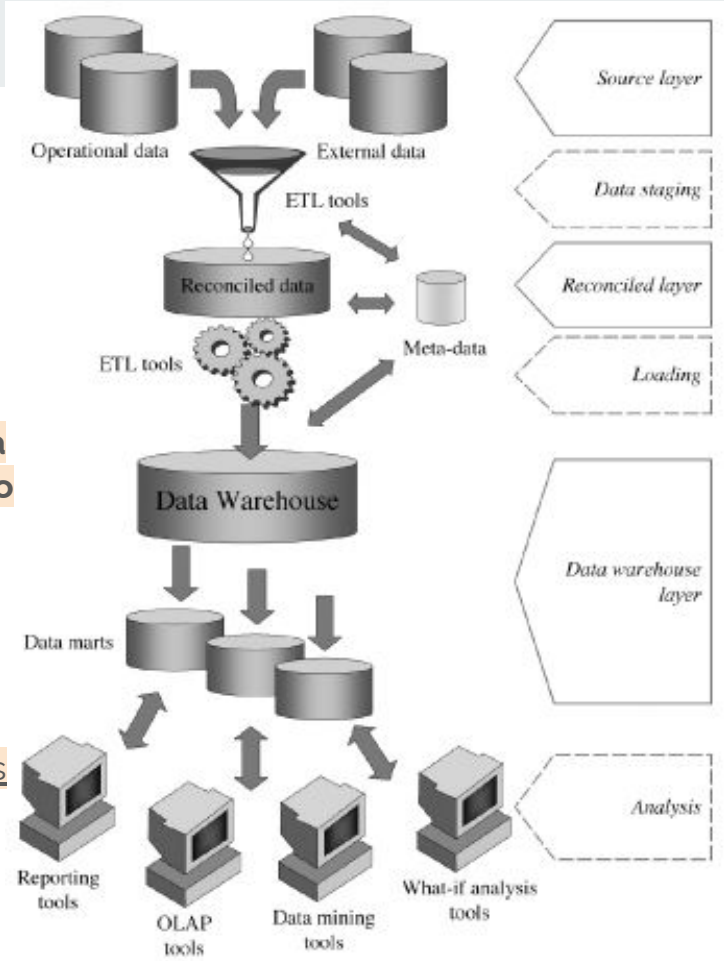


Figura 2.3: Almacén de datos. Arquitectura de tres capas.

Almacenes orientadas a la estructura

Arquitectura de tres capas: La arquitectura de tres capas agrega una capa intermedia llamada "capa de datos reconciliados" o "capa de integración de datos". Los datos se procesan y limpian en esta capa antes de ser transferidos al almacén de datos final.

- Se utiliza en grandes organizaciones que requieren consolidar datos provenientes de múltiples fuentes y necesitan mantener la integridad de los datos durante todo el proceso.
- Ejemplo: Una corporación con múltiples filiales en diferentes países. Los datos de cada país se integran y validan en la capa de reconciliación antes de almacenarse en el almacén de datos central. Esto permite generar informes consolidados a nivel global y local con consistencia.

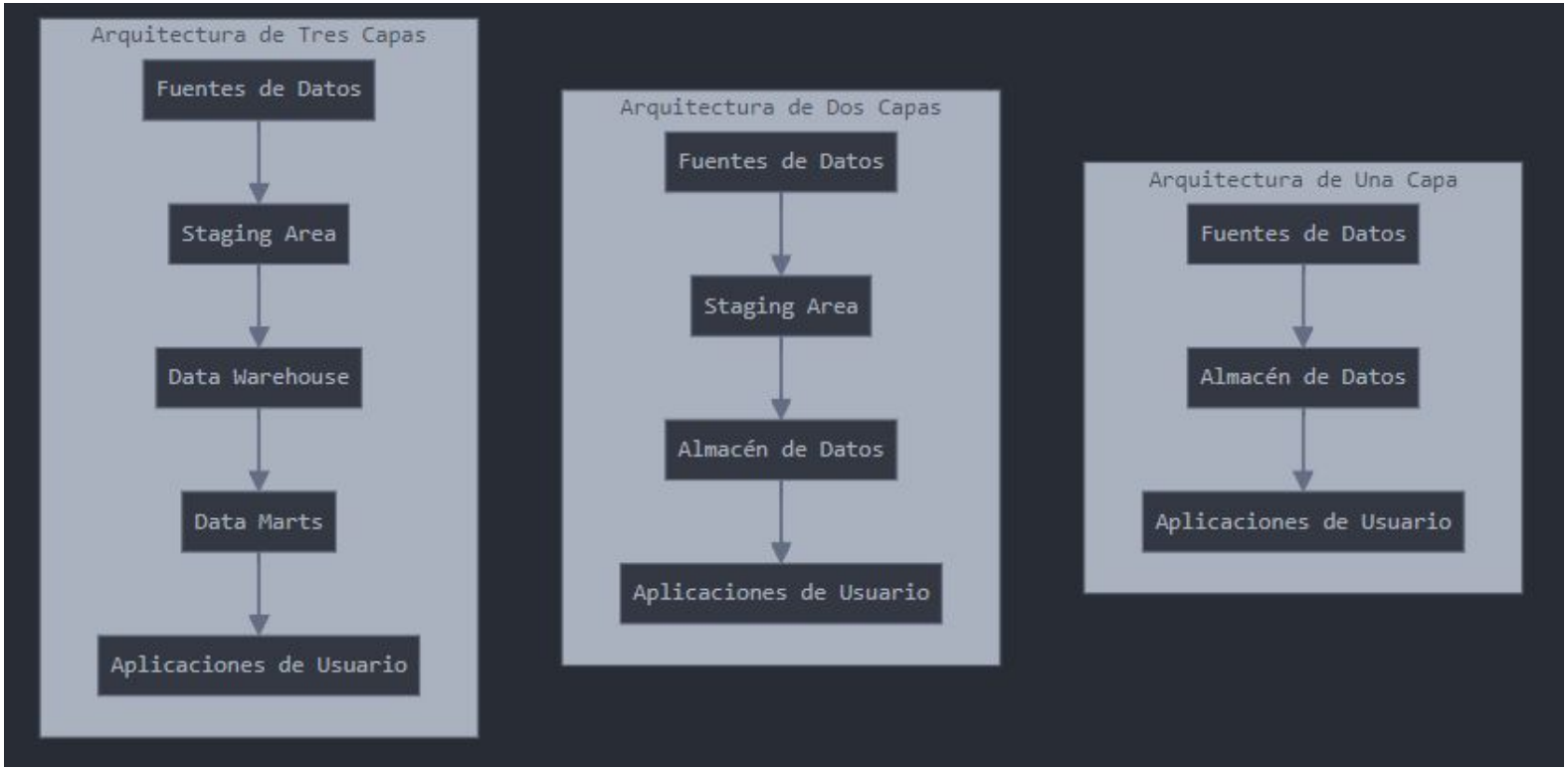
Ventajas:

- Mayor flexibilidad para incorporar nuevas fuentes de datos.
- Facilita la limpieza y estandarización de los datos antes de que lleguen al almacén principal.
- Reduce el riesgo de duplicados y errores en el almacén de datos.

Desventajas:

- Mayor complejidad de implementación y mantenimiento.
- Requiere herramientas y recursos avanzados para la integración y validación de datos.

Almacenes orientadas a la estructura



Importancia de Separar los Datos Transaccionales de los Estratégicos

La separación de datos transaccionales (operacionales) y datos estratégicos (analíticos) es fundamental para optimizar el rendimiento del análisis y asegurar la integridad de los procesos en entornos de Big Data. A continuación, se desarrollan las razones y beneficios de esta separación:

1. **Optimización de Consultas y Análisis**
2. **Mejora en la Calidad y Consistencia de los Datos**
3. **Seguridad y Control de Acceso**
4. **Soporte a Diferentes Tipos de Consultas**
5. **Reducción de Costos Operacionales**

Importancia de Separar los Datos Transaccionales de los Estratégicos

Importancia de Separar los Datos Transaccionales de los Estratégicos

Razones y beneficios de esta separación:

Optimización de Consultas y Análisis:

- **Los datos transaccionales están diseñados para soportar operaciones rápidas, como inserciones y actualizaciones en tiempo real.** Sin embargo, realizar análisis complejos (consultas OLAP) en estos datos puede reducir significativamente el rendimiento del sistema, ya que se sobrecarga la base de datos operacional.
- **La separación permite que las bases de datos analíticas (OLAP) se optimicen exclusivamente para consultas intensivas,** como agregaciones, sumas y análisis multidimensionales, sin impactar en las operaciones cotidianas.

Mejora en la Calidad y Consistencia de los Datos:

- **Los datos transaccionales suelen estar en constante cambio debido a operaciones diarias** (compras, ventas, registros de usuarios, etc.). Si se intentan analizar directamente en su entorno original, pueden surgir problemas de inconsistencia.
- **Al mover los datos a un almacén analítico, estos se "congelan" en un estado consistente y se someten a procesos de limpieza y validación**. Esto asegura que las decisiones estratégicas se basen en información precisa.

Importancia de Separar los Datos Transaccionales de los Estratégicos

Importancia de Separar los Datos Transaccionales de los Estratégicos

Razones y beneficios de esta separación:

Seguridad y Control de Acceso:

- **Los datos transaccionales generalmente incluyen información sensible que debe ser protegida (datos de clientes, información financiera, etc.).** Exponer estos datos a procesos de análisis puede aumentar el riesgo de violaciones de seguridad.
- **Separar los datos en capas transaccionales y analíticas permite aplicar políticas de seguridad específicas para cada entorno, garantizando que solo se acceda a la información estratégica en la capa analítica.**

Soporte a Diferentes Tipos de Consultas:

- **Las bases de datos transaccionales (OLTP) son eficientes para operaciones individuales (por ejemplo, actualizar un registro de cliente) pero no para consultas agregadas y reportes complejos.**
- Las bases de datos analíticas (OLAP) están optimizadas para trabajar con grandes volúmenes de datos históricos, permitiendo análisis longitudinales, minería de datos y visualización de patrones.

Importancia de Separar los Datos Transaccionales de los Estratégicos

Importancia de Separar los Datos Transaccionales de los Estratégicos

Razones y beneficios de esta separación:

Reducción de Costos Operacionales:

- Al separar las capas transaccionales y analíticas, las empresas pueden escalar cada sistema de manera independiente, evitando pagar por almacenamiento y procesamiento innecesario en la capa transaccional.
- Por ejemplo, el almacenamiento en un Data Lake puede ser mucho más barato para datos históricos que no se consultan frecuentemente.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. NoSQL.

Las NoSQL se caracterizan principalmente por:

Independencia del esquema: Al contrario que en las bases de datos relacionales, no es necesario diseñar un esquema para definir los tipos y estructura de los datos almacenados, permitiendo acortar el tiempo de desarrollo y facilitando las modificaciones de la estructura interna de la base de datos.

No relacionales: El concepto de relación de las bases de datos relacionales no existe en NoSQL. Por tanto, se trabaja con datos que no están normalizados, lo cual aporta flexibilidad en relación a los tipos y estructuras de datos que pueden ser almacenados.

Distribuidas: La cantidad de datos almacenados requiere de su almacenamiento en múltiples servidores, ya que un único servidor por potente que sea no podrá procesar en tiempos razonables tal cantidad de información. Este hecho permite utilizar hardware sencillo, ya que al utilizar múltiples servidores no es necesario que todos ellos tengan grandes prestaciones.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. NoSQL.

Las bases de datos documentales trabajan con documentos, entendidos como una estructura jerárquica de datos que, a su vez, puede contener subestructuras. Las bases de datos documentales pueden, efectivamente, trabajar con estos tipos de documentos. Sin embargo, el término documento en este contexto posee un mayor nivel de abstracción.

Los documentos pueden consistir en datos binarios o texto plano. Es posible que se traten de datos semiestructurados, cuando aparecen en formatos como JavaScript Object Notation (JSON) o Extensible Markup Language (XML). Por último, también pueden ser datos estructurados conforme a un modelo de datos particular como, por ejemplo, XML Schema Definition (XSD).

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

Actualmente, **XML y JSON** son los formatos de intercambio de datos más utilizados en el desarrollo de aplicaciones web.

XML:

- XML (Extensible Markup Language) surge como una extensión de SGML, un lenguaje de marcado genérico creado para definir gramáticas de lenguajes.
- XML permite definir reglas para codificar documentos con una sintaxis específica y etiquetas personalizadas. Se compone de elementos (las etiquetas que encierran datos) y atributos (datos adicionales dentro de las etiquetas).
- Su orientación a documentos lo hace muy flexible para representar información estructurada de forma jerárquica y compleja. Sin embargo, al centrarse en la estructura, XML tiene cierta redundancia y verbosidad.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

En XML, **la estructura principal de un documento está formada por dos elementos: el prólogo (opcional) y el cuerpo. El prólogo contiene a su vez dos partes: la declaración XML que establece la versión del lenguaje, el tipo de codificación y si se trata de un documento autónomo y la declaración del tipo de documento. El cuerpo, por su parte, contiene la información del documento.**

Supongamos que Carlos ha enviado un mensaje de whatsapp a Javier diciéndole que han quedado con los compañeros de trabajo a las diez de la noche en la puerta del Sol. Un documento XML que representa este mensaje como un documento, podría ser el mostrado en el listado.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

XML:

Listado 1: Quedada en la puerta del sol

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <whatsapp>
3         <para> Javier </para>
4         <de> Carlos </de>
5         <titulo> Quedada </titulo>
6         <contenido> A las 22:00 pm en la puerta del sol </contenido>
7 </whatsapp>
```

El listado 1 incluye en la primera línea el **prólogo del documento,** definiendo la versión y el tipo de codificación utilizada. A partir de la segunda línea, se define el cuerpo del documento que contiene, mediante etiquetas de apertura <> y cierre </>, los distintos atributos de los que se compone el documento.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

JSON:

- JSON (JavaScript Object Notation) nace como un subconjunto de la notación de objetos JavaScript, orientado al intercambio de datos entre aplicaciones web.
- Soporta menos tipos de datos que XML (cadenas, números, booleanos, arrays, objetos), pero tiene una sintaxis más simple y compacta basada en pares clave-valor.
- Al enfocarse solo en representar datos, JSON resulta más ligero y rápido de parsear que XML. Es ideal para el intercambio de información simple entre cliente y servidor.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

En JSON, la sintaxis del lenguaje tiene las mismas reglas que el lenguaje JavaScript, del cual proviene.

Los archivos JSON deben cumplir también otras reglas sintácticas adicionales. En primer lugar, un archivo JSON representará o bien un objeto, es decir, una tupla de pares clave-valor o bien una colección de elementos, es decir, un vector o array.

Los archivos JSON que representan objetos comienzan con una llave de inicio { y terminan con una llave de cierre }. Cuando se representa un vector, sus elementos se encierran entre corchetes []. Las cadenas y nombres de atributos del objeto deberán encerrarse entre comillas, así como todos los nombres de los atributos del objeto, separándose cada elemento del siguiente con una coma (,) no habiendo una coma después del último elemento.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

Así pues, si se pretende representar en formato JSON el mensaje que ha enviado Carlos a Javier, el fichero JSON resultante sería el mostrado en el listado 2.
Este fichero define un objeto JSON que contiene una serie de atributos entrecomillados cuyo valor asociado son cadenas de caracteres que, por tanto, también van entrecomilladas.

Listado 2: Quedada en la puerta del sol

```
1 {
2       "para": "Javier",
3       "de": "Carlos",
4       "titulo": "Quedada",
5       "contenido": "A las 22:00 pm en la puerta del sol"
6 }
```

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

Tabla 2: Tabla comparativa entre XML y JSON

	XML	JSON
Lenguaje fuente	SGML	JavaScript
Tipo Lenguaje	Orientado a datos	Orientado a documentos
Notación	Pesada	Ligera
Etiquetas inicio y fin	Sí	No
Comentarios	Sí	No
Espacios de nombres	Sí	No
Soporte tipos de datos	No	Sí

XML es más apropiado para documentos complejos con mucha estructura, mientras que JSON es mejor para transmitir objetos de datos simples de forma ágil y con menor sobrecarga. Ambos siguen siendo ampliamente utilizados como formatos universales de intercambio de información en el desarrollo web y de APIs.

Almacenes de datos. Bases de datos documentales u orientadas a documentos.

Actividad: Utilice la siguiente aplicación para convertir el código XML de ejemplo a JSON y conteste a las siguientes preguntas: XML to JSON Converter

- ¿Cuáles son las diferencias entre XML y JSON?
- ¿Cómo se representan los datos en XML?
- ¿Cómo se representan los datos en JSON?

XML

```
<producto>
 <nombre>Ordenador portátil</nombre>
 <precio>1.200 €</precio>
 <marca>Acer</marca>
 <modelo>Aspire 5</modelo>
 <caracteristicas>
  <caracteristica>Procesador Intel Core i5</caracteristica>
  <caracteristica>Memoria RAM de 8 GB</caracteristica>
  <caracteristica>Disco duro de 1 TB</caracteristica>
  <caracteristica>Pantalla de 15,6 pulgadas</caracteristica>
 </caracteristicas>
</producto>
```

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

Un grafo es un ente matemático compuesto por un conjunto de nodos o vértices y un conjunto de enlaces o aristas. Matemáticamente puede ser expresado por medio de la ecuación:

$$G = \{V,E\}$$

(Donde V representa el conjunto de nodos o vértices y E representa el conjunto de enlaces o aristas.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

El grafo de la siguiente figura representa un conjunto de ciudades conectadas por autovías, el conjunto V de nodos sería V = {La Coruña, Madrid, San Sebastián, Barcelona, Valencia, Sevilla, Cádiz}, mientras que el conjunto de enlaces E vendría dado por E = {A-1, A-2, A-3, A-4, A-4-I, A-4-II, A-6, A-7}.

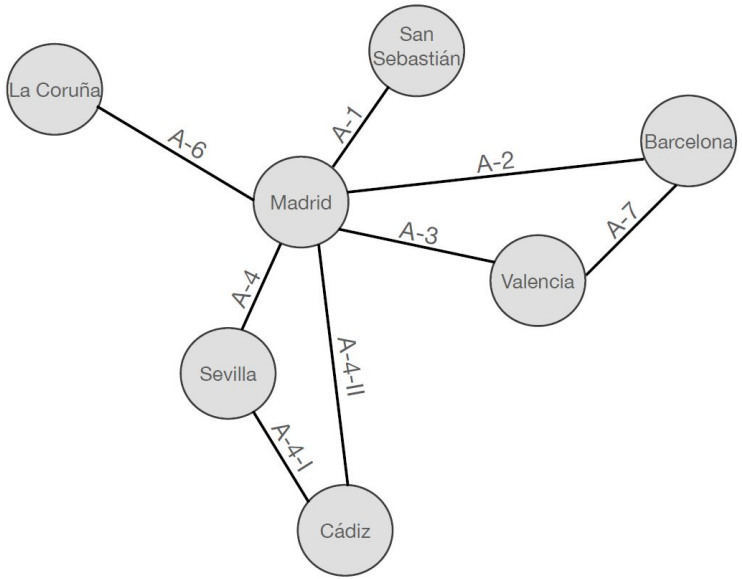


Figura 4: Grafo que representa las principales autovías de España

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

Una base de datos orientada a grafos es, por tanto, un sistema de bases de datos que implementa métodos de creación, lectura, actualización y eliminación de datos en un modelo expresado en forma de grafo.

Existen dos aspectos fundamentales en este tipo de sistemas: el primero de ellos hace referencia al almacenamiento de los datos. En una base de datos orientada a grafos, los datos pueden almacenarse siguiendo el modelo relacional, lo que implica mapear la estructura del grafo a una estructura relacional, o bien, almacenarse de forma nativa utilizando modelos de datos propios para almacenar estructuras de tipo grafo.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

La ventaja de mapear los grafos a una estructura relacional radica en que la gestión y consulta de los datos se realizará de forma tradicional a través de un backend conocido como, por ejemplo,
MySQL.

La ventaja del almacenamiento nativo de grafos radica en que existen modelos de datos e implementaciones que aseguran y garantizan el buen rendimiento y la escalabilidad del sistema.

El segundo aspecto importante es el procesamiento de los datos. El procesamiento nativo de los datos de grafos, el cual es beneficioso porque optimiza los recorridos del grafo cuando se realizan consultas aunque, en ocasiones, invierta demasiado tiempo y memoria en consultas que no requieren de recorridos complejos.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

Cualquier dominio puede ser modelado como un grafo. La motivación para requerir de sistemas de bases de datos específicos, orientados a trabajar con este tipo de datos, radica en tres aspectos principales:

- Rendimiento: Consultas más eficientes que se localizan en porciones del grafo, con complejidad constante al escalar. Superan a las BD relacionales.

- Flexibilidad: No requiere modelado exhaustivo previo. Se pueden agregar nodos y relaciones sobre la marcha sin modificar todo el modelo. Facilita la implementación.

- Agilidad: Gestión ágil de los datos gracias a las ventajas de rendimiento y flexibilidad. Permite metodologías de desarrollo ágil y diseño rápido de software que usa grafos.

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

Lenguajes de marcado de grafos:

- GraphML
- eXtensible Graph Markup and Modeling Language (XGMML)
- Graph Exchange Language (GXL)
- Graph Modelling Language (GML)

La mayoría de ellos son variantes o extensiones del lenguaje XML para el modelado de grafos.

GraphML es uno de los lenguajes más extendidos para el modelado de datos en forma de grafo. Se trata de un lenguaje sencillo, general, extensible y robusto. La notación es muy similar a XML. A modo de ejemplo, el grafo mostrado en la figura 4 podría representarse en GraphML según se muestra en el listado

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

GraphML

GraphML

Definición:

- Derivado de XML
- Muy extendido para el modelado de grafos (GML, GXL, XGMML)
- Sencillo
- General
- Extensible
- Robusto

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

- Se define un grafo llamado "Grafo_Autovias" con aristas no dirigidas (edgedefault="undirected"), por lo tanto los enlaces son bidireccionales.
- Se especifican los nodos del grafo, que representan ciudades.
- Se definen las aristas entre nodos, indicando el identificador de cada arista y sus nodos origen y destino.
- Al ser no dirigido, el origen y destino son intercambiables.
- Si fuera un grafo dirigido, se debería especificar edgedefault="directed" y origen/destino no serían intercambiables.
- Permite modelar la red de autovías entre distintas ciudades como un grafo no dirigido en GraphML.

Listado 3: Grafo Autovías

```
1 <graph id="Grafo_Autovias" edgedefault="undirected">
2       <node id="La Coruña"/> <node id="San Sebastián"/>
3       <node id="Madrid"/> <node id="Barcelona"/>
4       <node id="Valencia"/> <node id="Sevilla"/>
5       <node id="Cadiz"/>
6       <edge id = "A-6" source="La Coruña" target="Madrid"/>
7       <edge id = "A-1" source="Madrid" target="San Sebastián"/>
8       <edge id = "A-2" source="Madrid" target="Barcelona"/>
9       <edge id = "A-7" source="Barcelona" target="Valencia"/>
10      <edge id = "A-3" source="Madrid" target="Valencia"/>
11      <edge id = "A-4" source="Madrid" target="Sevilla"/>
12      <edge id = "A-4-I" source="Sevilla" target="Cádiz"/>
13      <edge id = "A-4-II" source="Madrid" target="Cádiz"/>
14 </graph>
```

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

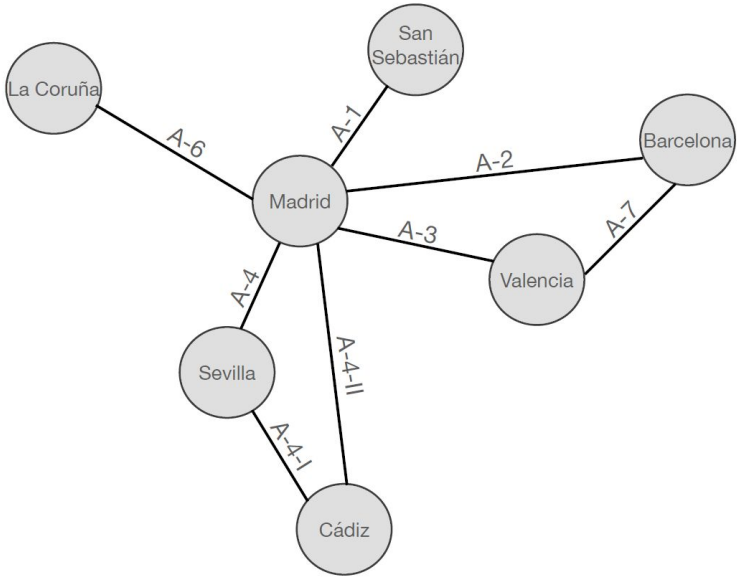


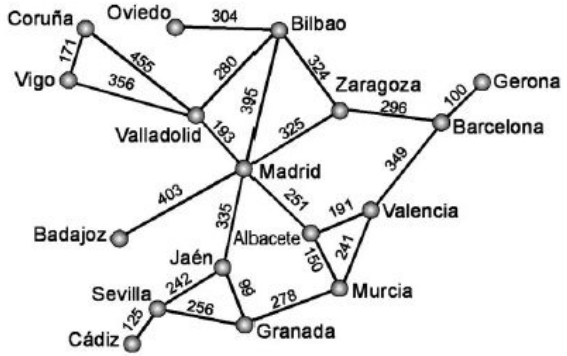
Figura 4: Grafo que representa las principales autovías de España

Listado 3: Grafo Autovías

```
1 <graph id="Grafo_Autovias" edgedefault="undirected">
2       <node id="La Coruña"/> <node id="San Sebastián"/>
3       <node id="Madrid"/> <node id="Barcelona"/>
4       <node id="Valencia"/> <node id="Sevilla"/>
5       <node id="Cadiz"/>
6       <edge id = "A-6" source="La Coruña" target="Madrid"/>
7       <edge id = "A-1" source="Madrid" target="San Sebastián"/>
8       <edge id = "A-2" source="Madrid" target="Barcelona"/>
9       <edge id = "A-7" source="Barcelona" target="Valencia"/>
10      <edge id = "A-3" source="Madrid" target="Valencia"/>
11      <edge id = "A-4" source="Madrid" target="Sevilla"/>
12      <edge id = "A-4-I" source="Sevilla" target="Cádiz"/>
13      <edge id = "A-4-II" source="Madrid" target="Cádiz"/>
14 </graph>
```

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

- Ente matemático
- Conjunto de vértices o nodos (V)
- Conjunto de aristas o enlaces (E)



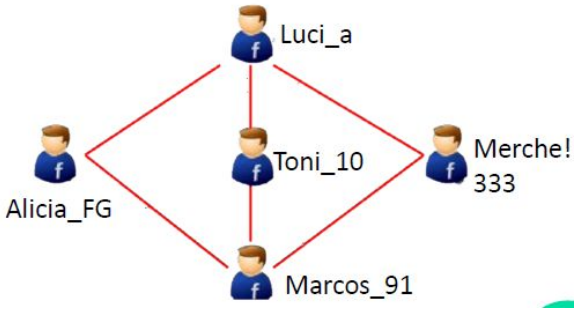
- Búsqueda de caminos mínimos
- Síntesis de circuitos
- Redes de comunicaciones
- Análisis de redes sociales
- ...

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos. Ejemplos

GraphML:

- |V| = 5
- |E| = 6
- Grafo no dirigido

Ejemplo: Facebook



GraphML:

Ejemplo: Facebook

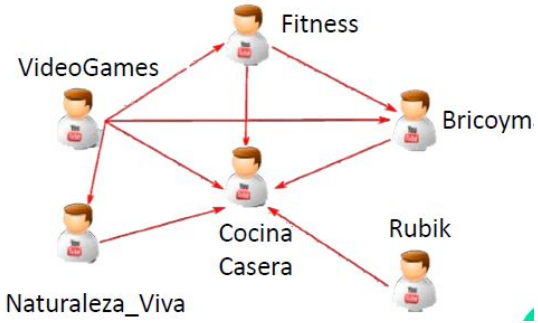
```
<graph id = "Grafo Facebook" edgedefault =
"undirected">
   <node id = "Luci_a"/>
   <node id = "Alicia_FG"/>
   …
   <edge id = "1" source = "Alicia_FG"
   target = "Luci_a"/>
   <edge id = "2" source = "Luci_a" target =
   "Toni_10"/>
   <edge id = "3" source = "Lucia_a" target
   = "Merche!333"/>
   <edge id = "4" source = "Alicia_FG"
   target = "Marcos_91"/>
   <edge id = "5" source = "Toni_10" target
   = "Marcos_91"/>
   <edge id = "6" source = "Merche!333"
   target = "Marcos_91"/>
</graph>
```

Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos. Ejemplos

GraphML:

- |V| = 6
- |E| = 9
- Grafo dirigido

Ejemplo: Youtube



GraphML:

Ejemplo: Youtube

```
<graph id = "Grafo Youtube" edgedefault =
"directed">
   <node id = "Fitness"/> <node id
   ="VideoGames"/>
   ...
   <edge id = "1" source = "VideoGames"
   target = "Fitness"/>
   <edge id = "2" source = "VideoGames"
   target = "Bricoymas"/>
   <edge id = "3" source = "VideoGames"
   target = "Cocina_Casera"/>
   <edge id = "4" source = "VideoGames"
   target = "Naturaleza_Viva"/>
   <edge id = "5" source = "Fitness" target
   = "Bricoymas"/>
   <edge id = "6" source = "Fitness" target
   = "Cocina_Casera"/>
   ...
</graph>
```

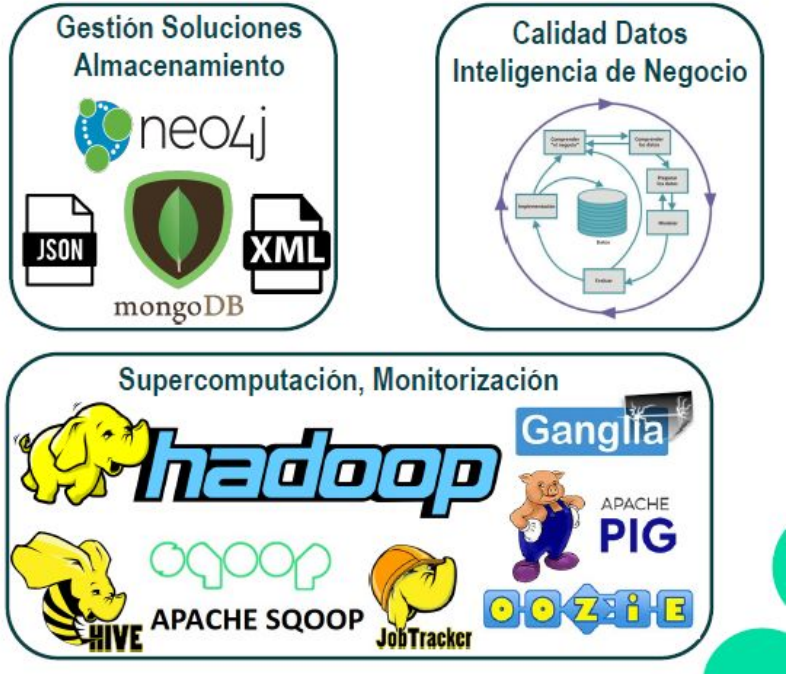
Almacenes de datos. Bases de datos documentales u orientadas a documentos. Grafos.

GraphML:

Visualización:

- yEd
- https://www.yworks.com/products/yed
- Gephi
- https://gephi.org

Tecnologías Big Data.

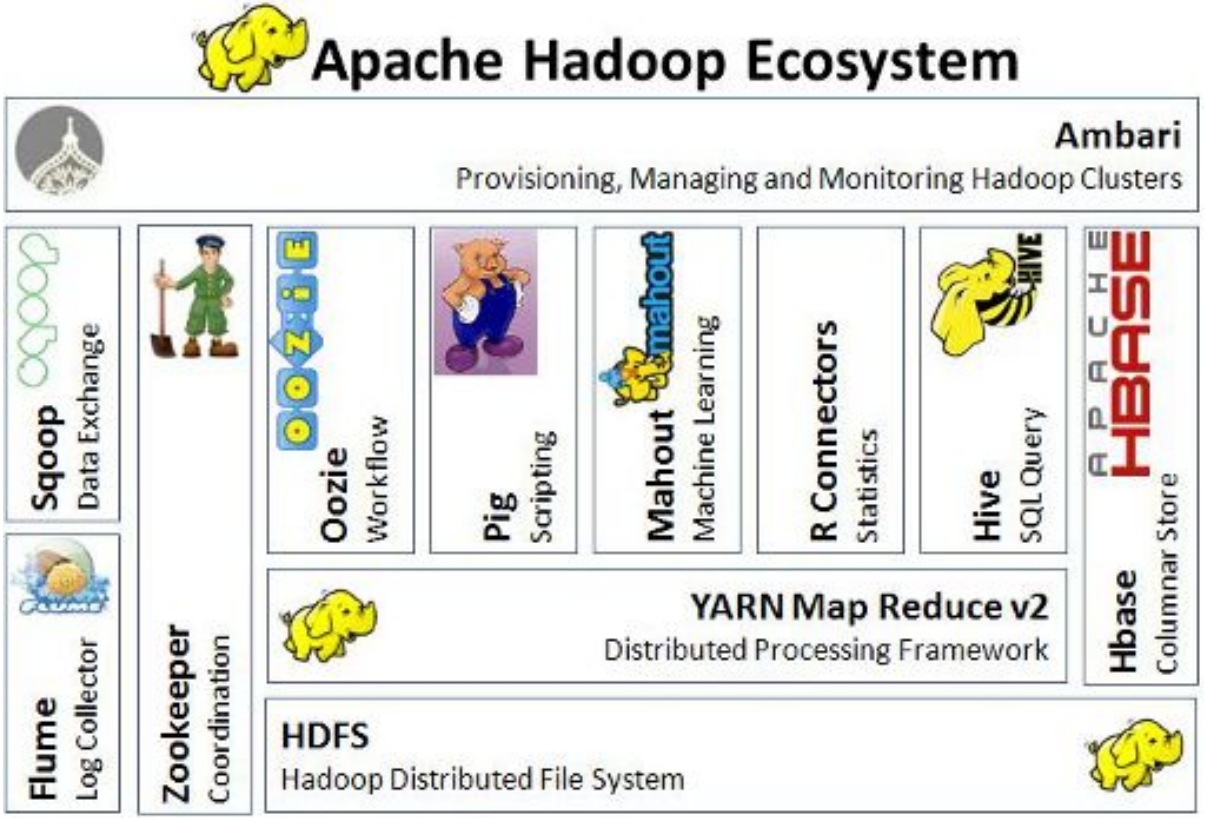


Tecnologías Big Data.

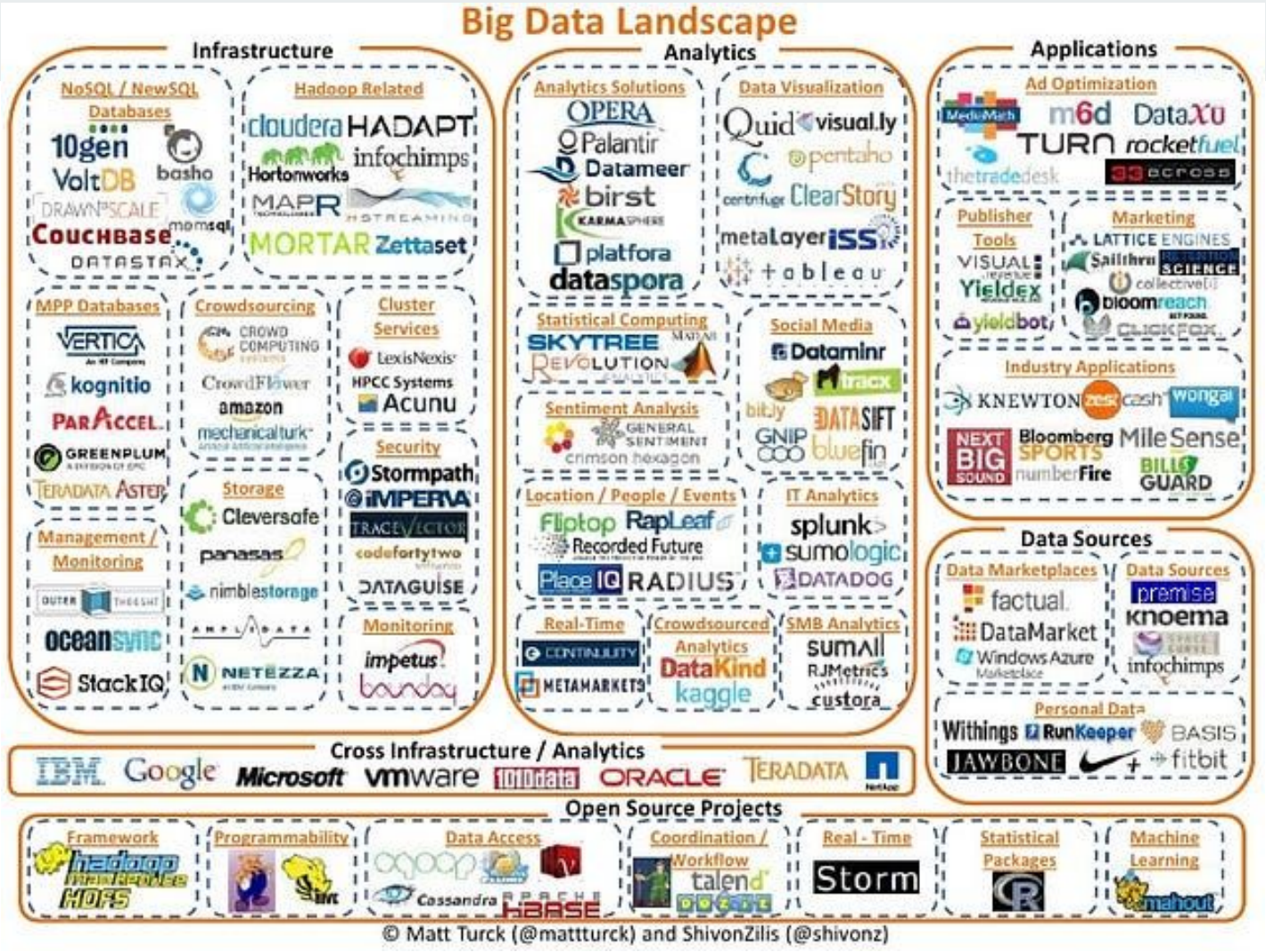
Toma de decisiones

Datos vs Tecnología

Tecnologías Big Data.

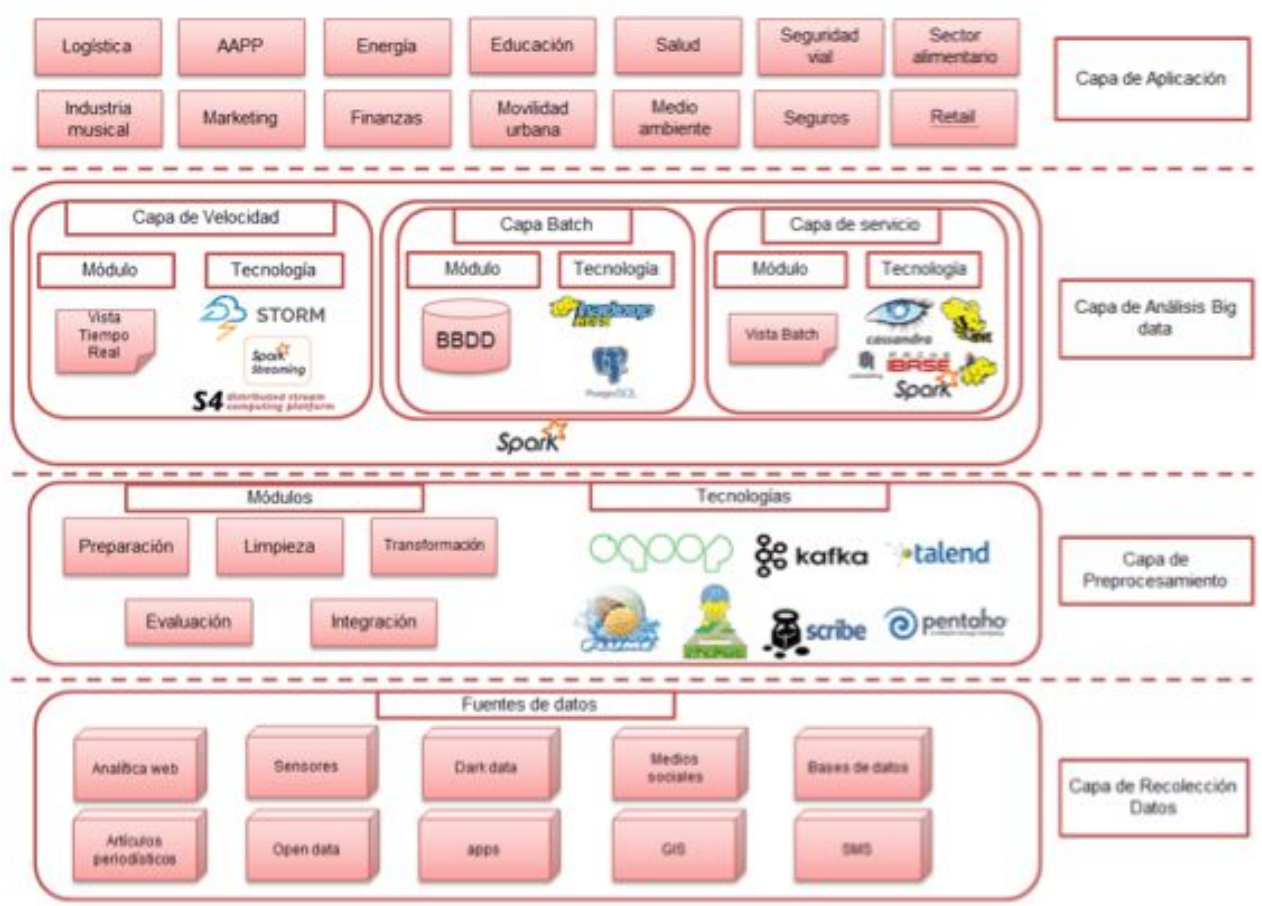


Tecnologías Big Data.



© Matt Turck (@mattturck) and ShivonZilis (@shivonz)

Capas arquitectura Big Data.



Ecosistema de herramientas, plataformas y soluciones de Big Data

- **Hadoop:** plataforma pionera para procesamiento distribuido de grandes volúmenes de datos. Incluye HDFS, MapReduce, YARN, Hive, HBase, entre muchos otros.
- **Spark:** motor de procesamiento en memoria altamente eficiente para trabajo batch y streaming. Compatible con ecosistema Hadoop.
- **Bases de datos NoSQL:** permiten escalar el almacenamiento y throughput, siendo aptas para datos no estructurados. Ejemplos: MongoDB, Cassandra, Redis.
- **Bases de datos NewSQL:** combinan escalabilidad de NoSQL con funcionalidades de SQL. Ejemplos: VoltDB, MemSQL.

Ecosistema de herramientas, plataformas y soluciones de Big Data

- **Lambda architecture:** combina procesamiento batch y tiempo real en capas separadas que se unen en una vista de servicios.
- **Kafka:** sistema de mensajería distribuida de altas prestaciones, suele usarse para streaming.
- **Flink:** plataforma de procesamiento de flujos de eventos para analytics en tiempo real y streaming.
- **Cloud computing:** servicios en la nube como AWS, GCP y Azure proveen recursos escalables para Big Data.

Ecosistema de herramientas, plataformas y soluciones de Big Data

- **Herramientas de ETL:** para extraer, transformar y cargar datos desde las fuentes. Ejemplo: Apache Nifi.

- **Machine learning:** algoritmos como regresión, árboles de decisión, redes neuronales. Se integran en pipelines.

- **Visualización:** Tableau, Power BI, Grafana para crear dashboards e informes a partir de los datos.

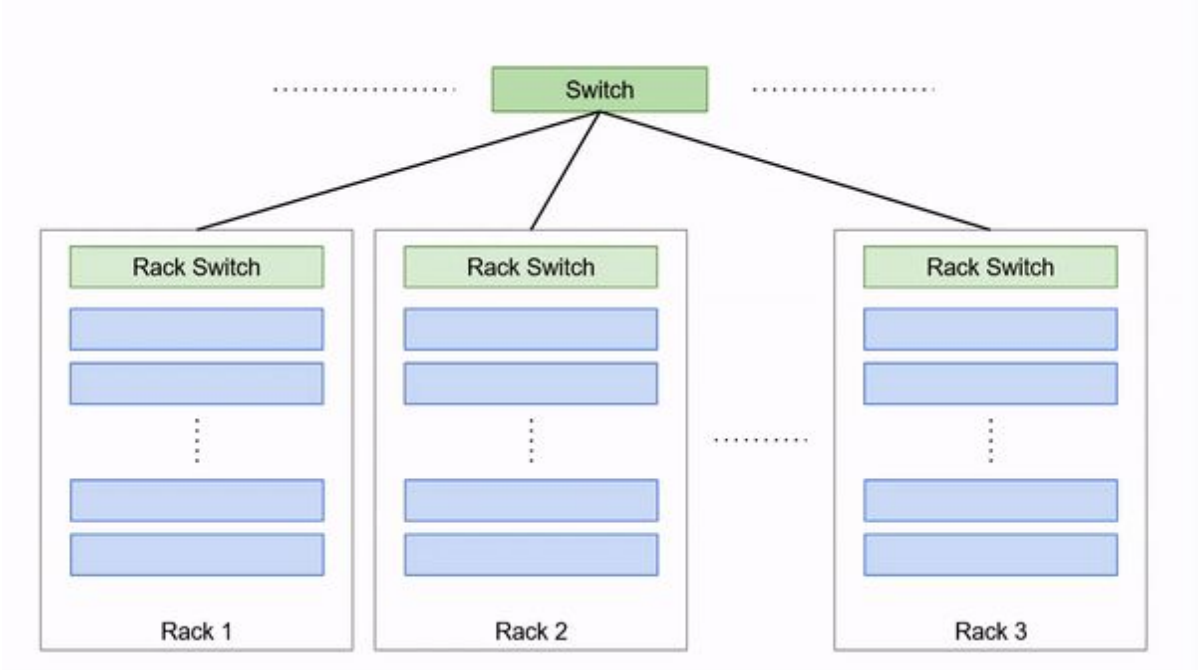
Ecosistema de herramientas, plataformas y soluciones de Big Data

Hadoop y su gestión de almacenamiento con HDFS (Hadoop Distributed File System)

- **HDFS** es el sistema de archivos distribuido que gestiona el almacenamiento de datos en Hadoop. Se encarga de almacenar grandes volúmenes de datos a través de múltiples nodos, distribuyendo los datos en bloques de tamaño predeterminado (generalmente 128 MB o 256 MB).
- **Principales características de HDFS**:
  - **Distribución de datos**: Los datos se dividen en bloques y se distribuyen en diferentes nodos para garantizar la disponibilidad y rendimiento.
  - **Replicación**: Cada bloque de datos se replica en múltiples nodos para asegurar la tolerancia a fallos.
  - **Gestión automática de fallos**: HDFS detecta y reequilibra los datos en caso de fallo de nodos.
- **Ventajas de HDFS en Big Data**:
  - Permite almacenar y procesar petabytes de datos a un bajo coste.
  - Alta tolerancia a fallos y escalabilidad.
  - Integra procesamiento masivo de datos con herramientas como **MapReduce** y **YARN.**

Ecosistema de herramientas, plataformas y soluciones de Big Data

Hadoop y su gestión de almacenamiento con HDFS (Hadoop Distributed File System)



Ecosistema de herramientas, plataformas y soluciones de Big Data

Comparativa de herramientas de almacenamiento masivo: Hive, Cassandra, etc.

Herramienta	Tipo de almacenamiento	Aplicaciones	Ventajas
Hive	Almacenamiento en tablas (HDFS)	Consultas SQL sobre datos masivos	Soporte SQL, fácil de usar
Cassandra	Columnar distribuido	Logs, transacciones	Alta escalabilidad y rendimiento
HBase	Columnar orientado a filas	Análisis en tiempo real	Compatible con HDFS
Amazon S3	Almacenamiento de objetos	Data Lakes, almacenamiento a gran escala	Bajo coste, integración en la nube
Google BigQuery	Almacenamiento masivo basado en columnas	Análisis y consultas rápidas	Consultas SQL a gran escala