INTRODUCCIÓN A BIG DATA

1. Porqué Big Data.

Debes saber que, para complementar la información que veremos en este curso sobre Big Data, existen multitud de fuentes online. La primera (y muy útil) referencia sería el artículo de la Wikipedia sobre lo que son los macrodatos (traducción al castellano de Big Data).

Macrodatos

Es igualmente muy importante que sepas que la literatura sobre Big Data (al igual que ocurre con las ciencias de la computación en general) es mas abundante en inglés que en castellano. Por ello, si quieres profundizar por completo por lo general tendrás que acceder a la versión en inglés de la información. Un ejemplo es el enlace equivalente al previamente indicado pero en la versión inglesa de la Wikipedia el cual contiene más información aún (de hecho si vas a optar por leer sólo uno de ellos, siempre opta por el artículo en inglés).

Big data (en inglés)

En esta sección hablaremos de las cinco características de Big Data que suelen emplearse para discernir si el procesamiento de datos que necesitamos realizar puede realmente considerarse como Big Data. A lo largo de la literatura existente acerca de Big Data,esas 5 características han venido en llamarse "las 5 Vs".

- Volumen.
- Velocidad.
- Variedad.
- Veracidad.
- Valor.

1. Volumen.

La primera característica del reto de tratamiento de datos que ha venido en llamarse Big Data es el volumen de los mismos, es decir, la gran cantidad de bytes de información que los componen. Podemos comprobar cómo ha ido creciendo en los últimos tiempos la cantidad de información almacenada por el ser humano, gracias a lo cual podemos hacernos una idea de la magnitud del reto del tratamiento de la información.

Hay que tener en cuenta que si bien el significado de un kilobyte (kB) es 1000 bytes, dado que en ambientes de computacionales se emplea constantemente la numeración en base 2 también existe el kibibyte (KiB), el cual corresponde a 1024 bytes (2^{10}). De igual modo,también existe el mebibyte (MiB = 2^{20} bytes), el gibibyte (GiB = 2^{30} bytes), y así toda la progresión hasta llegar al yobibyte (YiB = 2^{80} bytes).

Lo que puede producir algo de confusión es que por lo general se emplean las nomenclaturas en base 10 de forma indistinta para designar tanto a la de base 10 como a la(más o menos equivalente) de base 2. Es decir, cuando vemos 1 MB, es posible que signifique 10^6 bytes pero también es posible que signifique 2^{20} bytes. Puede depender tanto del fabricante del dispositivo como a qué se esté refiriendo (módulos de memoria ,unidades de almacenamiento, hardware de red, ...).

Para hacernos una idea del volumen de datos que maneja la humanidad, según las predicciones el volumen de datos en el mundo se calculaba en unos 4.4 zettabytes en 2013,y tiene un crecimiento

exponencial según el cual se espera que pueda llegar a los 163 zettabytes para el año 2025. **¿De dónde vienen todos esos datos?**

- Datos de usuarios y/o clientes de instituciones y empresas.
- Datos generados por transacciones (compras, transferencias, ...).
- Datos adquiridos por sensores (de temperatura, de humedad, ...).
- Datos subidos a redes sociales (textos, imágenes, vídeos, ...).
- Datos relacionados con la salud (historiales y pruebas realizadas a pacientes).
- Datos de geolocalización (posicionamiento en cada momento según GPS).
- Datos guardados en logs (de todos los accesos que hacemos a páginas web).
- Datos producidos por el Internet de las cosas (de los diversos dispositivos IoT).
- Datos producidos por la genómica (cada vez que se secuencia un genoma).
- Datos de meteorología (información obtenida por satélites y las predicciones realizadas a partir de la misma).
- Datos producidos por cámaras (imágenes estáticas y vídeos producidos).
- Datos producidos por micrófonos (grabaciones de sonido producidas).
- Datos de RFID (aquellos con los que se tratan al realizar identificación por radiofrecuencia).
- Datos producidos por los sectores energético e industrial (toda la información que se genera alrededor de la energía y la industria).
- Datos Open Data (todos los datos abiertos liberados ya sea a nivel gubernamental o no gubernamental).

¿A partir de qué cantidad de datos es Big Data?

No existe ninguna entidad u organismo que regule de algún modo cuál es el tamaño de datos concreto a partir de la cual se considera que estamos en un ambiente Big Data.

Simplemente nos quedaremos con que los sistemas para Big Data hoy en día trabajan con volúmenes del orden de los petabytes (PB) e incluso de los exabytes (EB).

Para saber más

En el siguiente enlace puedes ver más información los Datos Abiertos (o OpenData) sobre una de las fuentes de datos más interesantes, ya que son de dominio público.

Datos abiertos

2. Velocidad.

No sólo tratamos con una gran cantidad de datos que hay que almacenar y procesar, sino que tales datos a su vez se siguen produciendo a una gran velocidad.

Para hacernos una idea, se calcula que en el mundo se generan cada 60 segundos: 350.000 tweets. 300 horas de vídeo subidos a YouTube (más los que se suban a otras plataformas). 171 millones de correos electrónicos. 330 GBs de información generados por sensores de motores de aviones comerciales.

Si volvemos a revisar la enumeración de posibles fuentes de las que provienen los datos que vimos en el apartado anterior, y tenemos en cuenta que en el mundo hay del orden de 7870 millones de personas (datos de 2022), podremos seguir haciéndonos una idea de la gran velocidad a la que todos esos datos se siguen generando cada minuto que pasa.

El problema con respecto a la velocidad no es únicamente el hecho de que el volumen de datos continúe creciendo sin parar (ya que si hemos dimensionado el almacenamiento para el doble de lo que

necesitamos entonces aún quedará mucho tiempo para que tal tamaño de almacenamiento sea un problema), sino lo rápido que es necesario obtenerlos y ser capaces de integrarlos junto con los que ya tenemos.

De la gran velocidad a la que llegan datos nuevos nacen las estrategias de procesamiento tipo streaming, las cuales estudiaremos más adelante.

3. Variedad.

Además de tener que procesar una gran cantidad de datos que se generan cada vez más rápido, existe el problema añadido de la gran variedad existente en cuanto a la representación de tal información.

Datos estructurados: Los existentes en registros (filas) de bases de datos (típicamente relacionales), los cuales existen dentro de tablas con un esquema definido que nos indica de qué tipo de datos es cada una de las columnas (entero, decimal, textual, fecha, ...).

Datos no estructurados: Aquellos que no están regidos por un esquema. Por ejemplo:

- Vídeos.
- Imágenes.
- Audios.

Hay que tener en cuenta que la proporción de datos en el mundo que son no estructurados se estima en más de un 80% del total, lo cual es fácilmente comprensible teniendo en cuenta la naturaleza de los mismos. Sólo hay que comparar el espacio de almacenamiento que ocupa un vídeo (típicamente varios megabytes) con el que ocupa un registro en una base de datos(típicamente varios bytes).

Datos semiestructurados: Son datos definidos según una cierta estructura pero que no tienen naturaleza relacional (es decir, no son registros de una tabla con un esquema determinado).Por lo general se almacena en ficheros de texto siguiendo un cierto formato preestablecido, de modo que se mantiene la flexibilidad que ofrece el fichero (para poder almacenar lo que sea necesario) a la vez que es posible determinar qué significa cada una de las porciones de información que se encuentran dentro del mismo.

Ejemplos de formato de fichero en los que se guardan datos semiestructurados:

- CSV.
- XML.
- JSON.

Metadatos: Los metadatos son datos extra (muchas veces generados de forma automática) que se guardan acerca de los propios datos para favorecer su interpretabilidad posterior.

Ejemplos de metadatos que pueden acompañar a los datos convencionales son:

- Información extra sobre su estructura.
- Fuente.
- Autor.
- Fecha de creación.
- Resolución en pixels (si se trata de una imagen o un vídeo).
- Duración (si se trata de un vídeo).
- Frecuencia de muestreo (si se trata de un audio).
- Tipo de compresión.

4. Veracidad.

Un problema extra con el que tenemos que tratar es el hecho de que los datos no siempre cuentan con la calidad deseada o no son totalmente fieles a la realidad. Este término está muy relacionado con el concepto de relación señal/ruido en cualquier flujo de información.

- El ruido son datos que no pueden ser convertidos en información (ya sea porque no la contienen o porque ésta está corrupta y es irrecuperable).

- La señal está constituida por datos que sí pueden ser convertidos en información con sentido.

En el siguiente enlace de la Wikipedia puedes saber más sobre lo que significa la relación señal/ruido:

Relación señal/ruido

Por ello, por un lado es necesario conocer en qué condiciones se adquirieron los datos (para poder así estimar su nivel de veracidad), mientras que por otro lado en muchos casos será necesario llevar a cabo un procesamiento específico de los mismos con el fin de resolver posibles problemas y eliminar información inválida.

Por lo general los datos producidos de modo automático (como la generada cuando realizamos transacciones) contienen menos ruido que los que producen personas (como los posts de un blog).

5. Valor.

El concepto de valor en relación a los datos tiene que ver con cómo de útiles son estos para una institución, empresa o persona.

Tiene mucho que ver con el concepto de veracidad que ya hemos visto, ya que por lo general cuanto más veraces (fieles a la realidad) sean los datos, más valor se puede obtener de ellos.

También depende en gran medida del tiempo transcurrido desde que se produjeron tales datos. Por ejemplo, si estamos operando en bolsa, el dato que nos indica el valor de una acción es mucho más valioso si corresponde a hace 1 segundo que si corresponde a hace 1 hora. En términos generales, cuanto más rápido seamos capaces de hacer llegar el dato desde donde se produce al lugar en el que se toman las decisiones, más valor podremos obtener de ellos.

Es también muy importante que los datos sean lo más completos posible para poder producir el valor deseado. Es decir, no sólo que sean veraces (que lo que viene sea correcto) sino que sean completos (que venga todo lo que necesitamos).

Por último, la propia interpretación del dato también juega un papel vital a la hora de poder obtener valor. Por ejemplo sería absurdo estar almacenando un valor de temperatura obtenido por un sensor que está bajo tierra y querer utilizarlo para una científica sobre temperatura ambiente. En este caso el dato podría ser perfectamente veraz pero la interpretación del mismo estaría siendo errónea, lo cual disminuiría el valor producido.

2. Qué conseguimos gracias a Big Data.

En esta sección veremos qué nos aporta no sólo el ser capaces de obtener y almacenar con grandes cantidades de datos sino también el poder tratarlos y analizarlos gracias a las metodologías y tecnologías de Big Data.

Aportes generales de Big Data

Las metodologías y tecnologías para Big Data nos permiten realizar diversas operaciones con grandes cantidades de datos, entre las cuales se encuentran:

- Capturarlos desde sus orígenes.
- Integrarlos para poderlos almacenar de un modo unificado.
- Almacenarlos de un modo distribuido y replicado, gracias lo cual conseguimos altos valores de disponibilidad.
- Tratarlos de forma distribuida, empleando para ello un alto número de máquinas que los procesan en paralelo.
- Aplicar técnicas de minería de datos (también llamado ciencia de datos cuando esa minería de datos se realiza en ambientes Big Data) para crear modelos predictivos.
- Usar esos modelos para realizar predicciones a utilizar en sistemas automáticos.
- Crear visualizaciones y cuadros de mando usando tanto los propios datos como los modelos creados para así dar soporte a la toma de decisiones.

El ser capaces de realizar tales operaciones con los datos, nos permiten obtener los siguientes aportes y beneficios (entre otros):

- Generar registros más detallados mediante la integración desde diversas fuentes.
- Optimizar las operaciones de instituciones y empresas.
- Poder actuar de modo inteligente basándonos en la evidencia de los datos.
- Identificar nuevos mercados.
- Realizar predicciones basándonos en modelos creados a partir de los datos.
- Detectar casos de fraude e impagos.
- Dar soporte a la toma de decisiones.
- Realizar descubrimientos científicos.
- Ayudar a los médicos a detectar enfermedades en función del historial de los pacientes y las pruebas que se les realizan.
- Crear nuevos fármacos más efectivos y con menos efectos secundarios.

2.1. Desde los eventos al valor.

El tratamiento de los datos a lo largo de diversas capas de procesamiento sucesivas nos permite llegar desde los meros eventos que se producen en nuestro mundo hasta la sabiduría que necesitamos para obtener valor gracias a poder tomar las mejores decisiones.

Eventos:

En nuestro mundo se producen eventos constantemente.

- Una estación meteorológica, mediciones de temperatura, humedad, presión atmosférica, etc.
- Una cámara toma imágenes dentro de una fábrica.
- Alguien pide un préstamo.
- Alguien realiza una llamada telefónica.
- Alguien realiza un pago con tarjeta.
- Un hospital realiza una prueba médica a un paciente.…

Datos:

Los eventos son reflejados de algún modo, generándose de ese modo datos que pueden ser almacenados para su uso posterior.

- Registros de bases de datos.
- Ficheros (en diversos posibles formatos).

Información:

Cuando les damos contexto a los datos, organizándolos de algún modo lógico,tenemos información.

- Distintos registros referentes a pagos con tarjeta quedan almacenados en una misma tabla.
- Usamos jerarquías de carpetas para organizar distintos ficheros en función de su tipo o significado (fotografías, audios, facturas, ...).

Conocimiento:

Si tratamos la información dándole un significado, podemos obtener conocimiento.

- A partir de gran cantidad de datos se generan modelos mediante los cuales se representa la realidad y que pueden ser utilizados para realizar predicciones.

Sabiduría:

Si una vez tenemos conocimiento en forma de modelos predictivos añadimos el entendimiento necesario para saber de qué modo emplearlos. Como resultado obtenemos sabiduría.

Valor:

La sabiduría de por sí misma no genera ninguna acción. Sin embargo, si realizamos acciones basándonos en la sabiduría, esas acciones serán mejores que las que podamos tomar sin basarnos en los datos.

La diferencia entre el resultado que podemos obtener basándonos en la sabiduría que producen los datos y el que obtendríamos si no los hubiésemos tenido en cuenta para nada, es el valor añadido que conseguimos.

A pesar de que dentro de las tecnologías de Big Data se suele englobar lo relacionado con obtener valor del dato, en la práctica son la minería de datos o la ciencia de datos las disciplinas que terminan de obtener el valor (haciendo uso de esas tecnologías).

La Minería de Datos es una rama de la Inteligencia Artificial que emplea técnicas de Aprendizaje Automático para obtener valor de los datos.

La Ciencia de Datos en el fondo es misma Minería de Datos pero haciendo énfasis en que se realiza en entornos de Big Data.

Sin embargo, puedes encontrar los términos **Minería de Datos y Ciencia de Datos** siendo empleados de forma equivalente (Minería de Datos en Big Data y Ciencia de Datos fuera de Big Data).

3. Clusters de computadoras

En ambientes de computación, un cluster es un conjunto de computadoras (también referenciados como servidores o como nodos) conectados entre sí mediante red para trabajar como una única unidad resolviendo cargas de trabajo de forma conjunta.

Históricamente los clusters se construían utilizando computadoras especializadas muy caras. Sin embargo, más adelante han ido apareciendo diversos frameworks o plataformas de computación

distribuida que emplean computadoras de uso común (el llamado commodity hardware), gracias al considerable aumento sus prestaciones.

Ley de Moore:

Según la Ley de Moore, cada aproximadamente 2 años se duplica el número de transistores en los nuevos procesadores que salen a la venta.

Puedes ver más información sobre la Ley de Moore y lo que significa en el siguiente enlace:

Ley de Moore

Puedes ver más información sobre los clusters de computadoras en el siguiente enlace:

Clúster de computadoras

El uso de clústers nos da una serie de ventajas respecto al uso de computadoras de forma individual:

- Alto rendimiento.
- Alta disponibilidad.
- Equilibrado de carga.
- Escalabilidad.

Alto rendimiento:

Dado que cada componente del cluster es una computadora completa, con sus propios recursos (procesador, memoria y almacenamiento), las cargas de trabajo susceptibles de paralelización pueden acelerarse en gran medida dividiéndolas en subtareas y distribuyéndolas para que sean ejecutadas en los distintos nodos.

Gracias a esto se pueden resolver problemas muy complejos que no sería posible resolver en un tiempo razonable en una máquina individual por muy potente que ésta sea.

Alta disponibilidad:

Mediante una continua monitorización entre los propios nodos del cluster, se puede detectar la no disponibilidad de un subconjunto de los mismos (ya sea por fallo eléctrico, por avería o por corte de las comunicaciones) y se pueden tomar medidas para que los servicios o datos que hay (o había) en esas máquinas sigan estando disponibles.

- Rearrancando un nodo caído o arrancando un nuevo nodo para suplirlo.
- Respondiendo las peticiones desde otro nodo del clúster que también contenga una réplica de esos datos.

Equilibrado de carga:

El equilibrado de carga (o también balance o balanceo) se consigue mediante algoritmos destinados a distribuir las cargas de trabajo entre los diversos nodos del clúster para así evitar cuellos de botella. Tales cuellos de botella se producen cuando el envío de trabajos a nodos sobrecargados aumenta la latencia media con la que tales trabajos son finalizados.

Para ello, se realiza una monitorización del estado de carga de cada nodo y se decide para cada paquete de trabajo a qué nodo enviarlo, atendiendo a:

- El tamaño del trabajo.
- El estado de carga de cada nodo.
- La potencia de procesamiento de cada nodo.

Escalabilidad:

Gracias a que el clúster está formado por un número indeterminado de nodos, no sólo conseguimos una mayor potencia de cálculo al utilizarlos para una misma tarea, sino que podemos hacer crecer dicha potencia de cálculo añadiendo nuevos nodos. En otras palabras, la potencia de cálculo del clúster es ampliable.

Esta característica es muy desable para sistemas Big Data, ya que desaparece la necesidad de realizar una estimación de potencia necesaria a priori, lo cual en por lo general siempre lleva a una sobre estimación para guardar un margen de seguridad. Con un clúster escalable podemos comenzar con un número determinado de nodos e ir añadiendo más según sea necesario.

Es interesante conocer la diferencia entre escalado horizontal y vertical:

Escalado vertical (scale-in):

Es el que se consigue mejorando las características hardware de la computadora (individual) en el que se están ejecutando las cargas de trabajo (procesador, memoria o almacenamiento). Por lo tanto, está limitado por la mejor especificación de hardware que sea posible encontrar en el mercado. Por ello, aunque reciba el nombre de "escalado" en la práctica no sirve para conseguir la característica de escalabilidad.

Escalado horizontal (scale-out):

Es el que se consigue añadiendo más nodos a un clúster. Por ello es el tipo de escalado que realmente nos permite conseguir la característica de escalabilidad.

En los siguientes enlaces puedes ver más información sobre lo que significa alto rendimiento, alta disponibilidad, equilibrado de carga y escalabilidad:

Clúster de alto rendimiento

Clúster de alta disponibilidad

Equilibrio de carga

Escalabilidad

4. Conceptos de almacenamiento dedatos.

En esta sección vamos a realizar un recorrido a lo largo de una serie de conceptos relacionados con almacenamiento que es importante conocer si vamos a trabajar en entornos Big Data.

Veremos aquí un esquema/resumen para que puedas tener una vista general:

Base de Datos Relacional:

El tipo de bases de datos más utilizado en el mundo (pero no escalable para Big Data).

Dataset:

Un conjunto de datos (quizás enorme).

Almacén de Datos:

Una sistema especial para almacenar datos (típicamente para analítica).

ACID:

Una serie de propiedades que deben cumplir las bases de datos que vayan a ser usadas para realizar transacciones.

Teorema CAP:

Un teorema acerca de las propiedades que podemos conseguir en una base de datos distribuida.

BASE:

Un principio de diseño de base de datos distribuidas.

4.1. Base de Datos Relacinal.

El lenguaje comúnmente empleado para interactuar con bases de datos relacionales es SQL. Puedes ver información sobre SQL en el siguiente enlace:

SQL

Una base de datos relacional es un almacén de información que almacena registros dentro de tablas.

Dichas tablas constan de filas (una por cada registro) y de columnas (los atributos de los que está compuesto cada registro).

Para cada tabla se define un esquema, en el cual se indica qué atributos tienen los registros de la misma y de qué tipo son (entero, decimal, texto, fecha, ...).

Gracias a la uniformidad de los datos que hay dentro de cada tabla, los motores de bases de datos relacionales pueden ofrecer un altísimo rendimiento a la hora de realizar búsquedas. Tales búsquedas pueden realizarse dentro de una única tabla o incluso afectando a varias tablas a la vez, a través de las relaciones existentes entre las mismas (de ahí "base de datos relacional").

Una de las características clave de las bases de datos relacionales para su alto rendimiento es su capacidad para generar índices sobre columnas de las tabla, gracias a los cuales se acelera tanto la búsqueda dentro de una tabla en particular como en enlazado de registros de distintas tablas que están relacionados.

En el siguiente enlace puedes ver más información sobre lo que es un sistema de gestión de bases de datos relacionales:

RDBMS

4.2. DataSet.

Un dataset es una colección de datos que guardan una cierta relación debido a la cual tiene sentido tratarlos juntos.

Ejemplos de dataset son:

- Una colección de tweets.
- Una colección de posts.
- Una colección de imágenes.
- Una serie de registros de base de datos relacional.
- Una serie de registros de base de datos no relacional.
- Una sucesión de medidas de una estación meteorológica.

Tal dataset a su vez puede estar almacenado en diversos formatos:

- Ficheros de texto plano (CSV, XML, JSON, o sin formato en particular).
- Tablas de base de datos.
- Ficheros multimedia (imagen, vídeo, audio, ...).

Puedes ver más información sobre lo que es un dataset (en castellano "conjunto de datos") en el siguiente enlace:

Conjunto de datos

4.3. Almacén de Datos

Un almacén de datos (del inglés, data warehouse) es un repositorio central de datos a nivel institucional o empresarial, dentro del cual se almacenan tanto datos actuales como históricos.

Se emplean tanto para inteligencia de negocio (BI) como para realizar consultas analíticas, por lo que además de tablas relacionales también suelen incluir en su interior subsistemas de tipo OLAP.

Por lo general, los datos que contienen son cargados periódicamente desde sus fuentes (por ejemplo sistemas SCM, ERP o CRM) mediante procesos de tipo ETL. Esto significa que la información que contienen es por general una instantánea del estado de los datos a cierta fecha, por lo que se trata de un almacenamiento que por lo general no se va a emplear para el uso de transacciones (sino para inteligencia de negocio o analítica, la cual es su función).

En el siguiente enlace puedes ver más información sobre lo que es un almacén de datos.

Almacén de datos

4.4. ACID

ACID es el principio fundamental de diseño por el cual se rigen las bases de datos que se crean para uso transaccional.

Está formado por 4 características de obligado cumplimiento, correspondiendo cada una de ellas a una de las letras del acrónimo:

- ✔ Atomicidad (atomicity).
- ✔ Consistencia (consistency).

- ✔ Aislamiento (isolation).
- ✔ Durabilidad (durability).

Este tipo de gestión realiza un control pesimista de la concurrencia, dando por hecho que cualquier problema que pueda ocurrir ocurrirá tarde o temprano por poco probable que sea (aplicando la Ley de Murphy).

Para conseguirlo, el motor de la base de datos bloquea registros individuales e incluso tablas completas en determinados momentos para asegurar que la consistencia se mantiene en todo momento.

Atomicidad:

La atomicidad implica que las operaciones realizadas sobre la base de datos o bien tienen éxito o fallan por completo (en tal caso dejando la base de dato sexactamente como estaba antes de comenzar la operación).

Por ejemplo, si una transacción consiste en insertar dos registros y la primera inserción es exitosa pero la segunda falla (por ejemplo porque el formato de un atributo no cumple con el esquema de la tabla), el motor de la base de datos no consolida la primera inserción.

Consistencia:

La consistencia nos asegura que la base de datos siempre es vista desde fuera en un estado consistente, comprobando siempre que los datos cumplen con los esquemas y restricciones de las tablas antes de escribirlos en ellas.

Gracias a ello, cualquier base de datos en estado consistente sigue en estado consistente tras una transacción exitosa.

Aislamiento:

El aislamiento nos asegura que los resultados de una transacción no son visibles por otras operaciones hasta que tal transacción haya sido completada.

Esto significa que si una transacción consiste en insertar 2 registros, ningún otro usuario o proceso podrá hacer una selección en la que aparezca sólo el primero de ellos (verá ambos si la transacción ha finalizado, o ninguno si aún no ha terminado).

Durabilidad:

La durabilidad nos asegura que los resultados de las escrituras (inserciones o actualizaciones) en la base de datos son permanentes. Esto implica que tales escrituras no se pierdan en el caso de que la máquina se apague tras realizar la transacción, lo cual se consigue persistiendo (guardando) la información en un sistema de almacenamiento no volátil. Quedaría por lo tanto descartada una base de datos que mantuviese la información únicamente en memoria.

Es importante tener en cuenta que los accesos a almacenamiento no volátiles son alrededor de 2 órdenes de magnitud (es decir, unas 100 veces) más lentos que los accesos a memoria, razón por la cual el hecho de mantener la característica de durabilidad es un limitante para la velocidad a la que pueden operar las bases de datos que cumplen con ACID.

Puedes ver más información sobre ACID en el siguiente enlace:

ACID

4.5. Teorema CAP

El teorema CAP (también conocido como conjetura de Brewer) establece que una base de datos distribuida sólo puede cumplir como máximo con 2 de las siguientes 3 propiedades:

- ✔ Consistencia (consistency).
- ✔ Disponibilidad (availability).
- ✔ Tolerancia a particionamiento (partition tolerance).

En otras palabras, según el teorema, nunca puede cumplirse C+A+P, sino que habrá que escoger siempre entre C+A, C+P o A+P a la hora de diseñar la base de datos distribuída.

Esto también implica que a la hora de seleccionar una base de datos distribuida tendremos en primer lugar que decidir de cuál de las 3 características estamos dispuestos a prescindir (C o A o P), y asegurarnos de que la base de datos cumple con las 2 características de las cuales no prescindimos.

Consistencia:

Cualquier lectura realizada (independientemente de sobre qué nodo se produzca) siempre muestra el estado posterior a la última escritura realizada (sobre cualquier nodo) o un error. Es decir, la base de datos tiene permitido devolver un error si no puede devolver el estado más actual, pero en ningún caso puede devolver un estado ya desfasado.

Disponibilidad:

Toda petición recibe una respuesta no errónea, sin la garantía de que el estado observado sea el correspondiente a la última escritura en algún nodo de la base de datos.

Tolerancia al particionamiento:

El sistema sigue funcionando y produciendo respuestas aún en el caso de que se haya perdido la comunicación con/entre algunos nodos, lo cual implica que se pueden recibir lecturas desde unos nodos que no incluyan información escrita en otros.

Para mostrar la razón por la que sólo 2 de las 3 propiedades del teorema CAP pueden cumplirse a la vez en una de base de datos distribuída, veremos los siguientes escenarios:

- ✔ Si se requiere consistencia (C) y disponibilidad (A), los nodos necesitan estar comunicados para asegurar la consistencia y poder devolver siempre respuestas que no sean de error. Por lo tanto asegurar la tolerancia al particionamiento (P) no es posible.
- ✔ Si se requiere consistencia (C) y tolerancia al particionamiento (P), los nodos no pueden mantenerse disponibles (A) durante el tiempo necesario hasta que termine el particionamiento y vuelva a alcanzarse un estado consistente (C) entre ellos.
- ✔ Si se requiere disponibilidad (A) y tolerancia al particionamiento (P), entonces la consistencia (C) no es posible debido a la necesidad de comunicación entre nodos para que ésta se mantenga. En otras palabras, si se quiere mantener disponible la base de datos en momentos de particionamiento, es obligatorio aceptar lecturas inconsistentes.

En el siguiente enlace podrás ver más información acerca del teorema CAP.

Teorema CAP

4.6. BASE

BASE es un principio de diseño de bases de datos basado en las restricciones impuestas por el teorema CAP, y típicamente empleado por muchas implementaciones de bases de datos distribuídas.

El significado del acrónimo es:

- ➔ Básicamente disponible (básically available).
- ➔ Estado blando (soft state).
- ➔ Consistencia eventual (eventual consistency).

Una base de datos que conforme a la filosofía BASE prefiere la disponibilidad antes que la consistencia (es decir, desde el punto de vista del teorema CAP es A+P).

Básicamente disponible:

Significa que la base de datos siempre responde a las solicitudes recibidas, ya sea con una respuesta exitosa o con una notificación de error, aún en el caso de que se produzca particionamiento entre los nodos (que algunos de ellos caigan o no están accesibles mediante red). En ocasiones eso puede significar recibir lecturas desde nodos que no han recibido la última escritura, por lo que el resultado puede no ser consistente.

Estado blando:

Implica que la base de datos puede encontrarse en un estado inconsistente cuando se produce una lectura, de modo que podemos realizar dos veces la misma lectura y obtener dos resultados distintos a pesar de que no haya habido ninguna escritura entre ambas.

En otras palabras, en cada momento sólo tenemos cierta probabilidad de estar viendo el estado final de la base de datos, porque puede haber escrituras que aún no se hayan consolidado en el nodo sobre el que se realiza la lectura.

Eventualmente consistente:

Significa que tras cada escritura, la consistencia de la base de datos sólo se alcanza una vez el cambio ha sido propagado a todos los nodos (de ahí que la consistencia sea eventual en lugar de segura). Durante el tiempo que tarda en producirse la consistencia, observamos un estado blando de la base de datos.

Puedes ver más información sobre la filosofía BASE en el siguiente enlace. Ten en cuenta que está en inglés ya que no había una versión en castellano en el momento de crear este documento.

Eventual consistency (en inglés)

5. Conceptos de procesamiento de datos.

En esta sección vamos a realizar un recorrido a lo largo de una serie de conceptos relacionados con procesamiento de datos que es importante conocer si vamos a trabajar en entornos Big Data.

Veremos aquí un esquema/resumen para que puedas tener una vista general:

- **Procesamiento en paralelo:**Distintos procesos dentro del mismo procesador.
- **Procesamiento en distribuido:**Distintos procesos para un mismo trabajo ejecutándose en distintas máquinas.

- **Estrategias de procesamiento de datos:**Cómo trabajamos con datos según el tipo de actividad que vayamos a realizar.
- **OLTP:**Procesamiento transaccional.
- **OLAP:**Procesamiento para analítica.
- **Principio SCV:**Un principio que nos dice qué propiedades podemos conseguir en un sistema de procesamiento distribuido.

5.1. Procesamiento en paralelo.

El procesamiento en paralelo tiene que ver con la capacidad de los sistemas operativos modernos (multitarea) de realizar varias tareas al mismo tiempo.

Los sistemas operativos multitarea existen desde mucho tiempo antes de que comenzasen a aparecer procesadores multihilo (con capacidad hardware para ejecutar varios hilos de forma concurrente) o las placas base multiprocesador (con capacidad para instalar varios procesadores). Para ello, cuentan con un gestor de procesos que se encarga de repartir el tiempo de ejecución entre los diversos procesos que estén ejecutándose en el sistema (en ventanas de tiempo de unos cuantos milisegundos).

Gracias a la aparición de placas base multiprocesador (que datan de tiempos previos a la aparición de los procesadores multihilo), los sistemas operativos instalados en las máquinas equipadas con una de tales placas pudieron tener hardware disponible como para poder realizar más de una tarea realmente al mismo tiempo.

Más adelante, con la aparición de los procesadores multihilo y (algo después) de los procesadores multinúcleo (con varios núcleos que a su vez por lo general también son multihilo), por fin la multitarea real pudo democratizarse para llegar al usuario convencional.

Multinúcleo:

El procesador contiene varios núcleos, cada uno de ellos con una CPU completa.

Multihilo:

Caso en el que una CPU está diseñada de modo que es capaz de atender a más de un hilo de ejecución (por lo general 2) permitiendo al segundo utilizar recursos que en ese momento no esté utilizando el primero.

Por ejemplo, el segundo proceso puede realizar una multiplicación de dos valores mientras el primero no está utilizando el recurso necesario para multiplicar porque se encuentra cargando un dato desde memoria.

Debido a que en ocasiones ambos procesos necesitan usar el mismo recurso (por ejemplo si ambos necesitan multiplicar en un determinado momento), habrá fracciones de tiempo en los que uno de ellos queda parado a la espera de que el otro termine de utilizar el recurso.

Por esa razón, un procesador comercializado como "de 2 núcleos y 4 hilos" no llega a ser equivalente a un procesador de 4 núcleos físicos.

Cuando las tareas que se están ejecutando son independientes (por ejemplo reproducir un audio, grabar vídeo con una webcam y ejecutar un editor de textos), no existe ningún problema de paralelización entre ellas.

El problema en cuanto a la paralelización aparece cuando tenemos una tarea muy compleja (por ejemplo un análisis sobre una gran cantidad de datos, lo cual es el típico ejemplo de trabajo en ambientes Big

Data), y necesitamos dividirla en distintas subtareas **independientes** (lo cual no siempre es posible, y aun siendo posible en ocasiones es muy complicado).

Tarea paralelizable:

Si nos piden sumar mil billones de números aprovechando la potencia de un procesador multinúcleo, podemos separar esos números en tantos paquetes como núcleos y ejecutar un proceso para cada uno de ellos.

Cada proceso realiza la suma de los números del paquete recibido y le devuelve el resultado a otro proceso al que ya sólo le queda sumar esos resultados parciales para obtener el resultado final.

Tarea no paralelizable:

Imaginemos que nos piden realizar la siguiente operación con mil billones de números (también aprovechando el procesador multinúcleo):

Comienza con un resultado de valor 0.
Mientras queden números, toma el siguiente y:
    Si valor actual del resultado es par, súmele el número.
    Si valor actual del resultado es impar, réstele el número.

Esta tarea presenta un evidente problema si queremos paralelizarla, ya que para cada nuevo paso necesitamos conocer el resultado parcial que se ha obtenido hasta el paso anterior.

Esto significa que no hay un modo eficiente de repartir el trabajo entre los núcleos. Aún separando los números en bloques consecutivos, el proceso que recibe el segundo bloque necesitará esperar a que el que recibe el primer bloque produzca su resultado, el que recibe el tercero al que recibe el segundo, y así sucesivamente.

5.2. Procesamiento en distribuido.

El procesamiento distribuido está muy relacionado con el procesamiento en paralelo, con la distinción de que en este caso el procesamiento se lleva a cabo en distintas máquinas que se comunican mediante red formando un clúster.

A su vez, cada una de las máquinas del cluster por lo general contará con un procesador multinúcleo, de modo que el procesamiento puede realizarse distribuido en nodos y a su vez en paralelo en procesador. Esto añade una complejidad extra al sistema a la hora de determinar el modo eficiente de aprovechar este doble nivel de paralelismo. Afortunadamente, gracias a los frameworks para Big Data existentes en el mercado, esta gestión se realiza de forma automática y por lo tanto transparente para el desarrollador.

Llegados a este punto, es importante tener en cuenta que para realizar su trabajo de forma conjunta, en muchas ocasiones es necesario que se produzca algún tipo de comunicación entre los distintos procesos que se están ejecutando en paralelo (muchas veces en la forma de un proceso entregando/enviando a otro un conjunto de datos con resultados parciales).Toda comunicación de datos necesita su tiempo, por lo que cuando se diseñan y usan sistemas Big Data hay que ser consciente de la diferencia en tiempo según dónde se ubiquen los procesos que se van a comunicar.

Comunicación dentro de la misma máquina:

Es el caso de comunicación más rápido posible, ya que los datos no necesitan ser enviados por red sino que los procesos pueden intercambiarlos a través recursos residentes en la propia máquina.Memoria

- RAM.
- Sistema de ficheros.
- Base de datos (útil pero lento).

Comunicación entre dos máquinas dentro del mismo switch:

La comunicación entre máquinas a través de una red de comunicaciones siempre es varias órdenes de magnitud más lenta que si se realiza entre procesos residentes en la misma máquina. Esto se debe a que es necesario enviarla al interfaz de red, este tiene que enviarla utilizando el protocolo de comunicaciones que corresponda, y la otra máquina tiene que leer la información recibida desde su propio interfaz de red.

Hay que tener en cuenta que, si bien es más lento, este tipo de comunicación es totalmente obligatoria en entornos de Big Data (los cuales precisamente se basan en el uso de distintos nodos dentro de un cluster).

Llegados al punto de necesitar comunicar datos entre máquinas, el modo más rápido, son los nodos conectados al mismo switch, ya que es el caso en el que la información necesita realizar el menor número de saltos entre elementos de la red.

Comuncación fuera del switch:

Cuantos más saltos entre switch interconectados necesite dar la información para llegar de un nodo a otro, más lenta será la comunicación.

El número de bocas de los switchs es limitado, por lo que típicamente todos los servidores dentro de un mismo rack en un CPD están conectados a un mismo switch, y hay una jerarquía de switches mediante la cual se comunica a unos racks con otros. El software que gestiona la infraestructura del clúster se configura con información acerca de la jerarquía que se ha utilizado a la hora de realizar esas conexiones físicas entre los nodos, de modo que pueda tomar las mejores decisiones a la hora de emplazar tanto datos como procesado en los nodos, con el fin de minimizar el número de veces que la información tiene que saltar entre switches.

Comunicación fuera del CPD:

Los nodos del cluster estarán típicamente todos dentro del mismo CPD, pero los datos no se originan dentro del CPD sino fuera de él, por lo que habrá que hacerlos llegar al mismo. De igual modo, los resultados producidos por el cluster en gran cantidad de casos son para ser consumidos o utilizados fuera de su CPD, por lo que habrá que sacarlos del él.

Esta comunicación se realiza a una velocidad muy inferior a la que se puede obtener a través de un switch, de modo que este es el tipo de comunicación más lento de todos.

5.3. Estrategias de procesamiento de datos.

En entornos Big Data se emplean distintas estrategias a la hora de procesar los datos, las cuales se escogen según la cantidad y naturaleza de los mismos, así como de las necesidades de la actividad que se esté realizando.

- ✔ Por lotes (del inglés batch).

- ✔ Transaccional.
- ✔ En tiempo real (del inglés realtime).
- ✔ Streaming (anglicismo usado sin traducir al castellano).

Por lotes:

El procesamiento por lotes (o también en inglés offline, por contraposición a online, que denota el de tiempo real), se realiza sin la necesidad de producir respuestas en un corto plazo. Pueden tardar en ejecutarse horas o incluso días.

Esta estrategia se emplea mayormente para trabajo de analítica con gran cantidad de datos (en ocasiones todos los disponibles para la tarea en cuestión).

Transaccional:

Al contrario del caso del procesamiento por lotes en el que el tiempo transcurrido hasta que se produce una respuesta no es demasiado importante, en el caso de las tareas transaccionales es de obligado cumplimiento que el tiempo necesario sea muy corto (a ser posible siempre por debajo de un segundo).

Este tipo de procesamiento se emplea cuando se realizan transacciones. Debido a restricciones en cuanto a tiempo de las tareas transaccionales, este tipo de procesamiento no puede afectar a un gran volumen de datos.

En tiempo real:

Al igual que en el caso del procesamiento transaccional, este tipo de procesamiento produce resultados en un corto espacio de tiempo.

Se emplea para analíticas interactivas (por lo general de tipo descriptivo), en las que un usuario humano está consultando estadísticas acerca de los datos (razón para que el tiempo de respuesta deba ser pequeño).

Para poder realizarse con un gran volumen de datos se suelen emplear subsistemas de tipo OLAP , en muchas ocasiones almacenadas en memoria.

Es importante destacar que en muchas ocasiones escucharemos decir que el procesamiento transaccional se produce en tiempo real, lo cual es totalmente cierto. En otras palabras, "tiempo real" puede en la práctica emplearse tanto para procesamiento analítico como para actividades transaccionales.

Al contrario no ocurre lo mismo. Es decir, no sería correcto decir que un procesamiento analítico en tiempo real es transaccional, ya que el término transaccional ya incluye la existencia de una transacción, lo cual no está ocurriendo cuando lo que estamos haciendo es analizar datos.

Streaming:

El procesamiento en streaming tiene mucho que ver con el que se produce en tiempo real en cuanto a que debe tener un corto tiempo de respuesta. Sin embargo, en este caso la clave es que ha de producirse a la velocidad a la que se recibe el flujo (de ahí streaming) de datos de entrada.

Esto añade una complejidad extra a los sistemas que han de diseñarse para ser capaces de procesar/analizar datos en streaming. Esto se debe a que las estructuras de datos en las que se mantiene la información necesaria para realizar la analítica deben ser capaces de actualizarse a

la medida que llegan nuevos datos. Por ello, necesitan almacenar esa información en memoria, lo que implica un tope máximo en el tamaño de datos que pueden tratarse a la vez.

5.4. OLTP.

Empleamos el acrónimo OLTP para designar un sistema que está orientado a transacciones, por lo que trata con los datos operacionales del día a día, relacionados con acciones que necesitan realizarse en tiempo real (de ahí que se llamen "online").

Tales transacciones son por lo general realizadas contra bases de datos relacionales, incluyendo sólo acciones sencillas (insertar, seleccionar, actualizar o eliminar) sin ningún tipo de analítica, gracias a lo cual se consiguen tiempos de respuesta inferiores a un segundo de modo que el sistema sea usable sin producir tiempos de espera desagradables para los usuarios.

En este enlace puedes ver más información acerca de lo que significa OLTP.

OLTP

5.5. OLAP.

Empleamos el acrónimo OLAP para designar un sistema que está orientado a procesar consultas de tipo analítico en tiempo real. Este tipo de sistemas son parte integral de la inteligencia de negocio (BI) y la minería de datos, usándose en todos los niveles de la analítica de datos (desde el análisis descriptivo hasta el prescriptivo).

Almacenan los datos en bases de datos multidimensionales (en ocasiones llamadas "cubos OLAP" debido a dicha estructura multidimensional), altamente optimizadas para poder responder en muy poco tiempo a consultas complejas que afectan a lo que en el mundo relacional/transaccional correspondería a varias tablas. Para obtener tal rendimiento, tales bases de datos multidimensionales guardan los datos denormalizados. En ocasiones se mantienen en la memoria RAM de la máquina que ejecuta el sistema, lo cual en tal caso implica un límite máximo respecto de la cantidad de datos que se pueden tratar a la vez.

Para entender qué significa que en OLAP se guarden los datos denormalizados, antes necesitas entender qué es la normalización de datos que se emplea en el mundo de las bases de datos relacionales.

Accede al siguiente enlace para ver más información:

Normalización de bases de datos

En el siguiente enlace puedes ver más información sobre lo que significa OLAP.

OLAP

5.6. Principio SCV.

Mientras que el teorema CAP tiene que ver con almacenamiento de datos distribuidos, el principio SCV está relacionado con el procesamiento distribuído de los datos. Es decir, no tiene que ver con la escritura y lectura (consistente o no) de los datos en entornos distribuídos sino con el procesamiento que se realiza sobre ellos dentro de los nodos de un sistema de procesamiento distribuído.

De modo similar a lo que ocurría con el teorema CAP, el principio SCV establece que un sistema de procesamiento distribuído sólo puede soportar como máximo 2 de las siguientes 3 características.

- Velocidad (speed).
- Consistencia (consistency).
- Volumen (volume).

Velocidad:

Se refiere a cuánto tardan en procesarse los datos desde el momento en el que son recibidos en el sistema analítico. Por lo general se excluye el tiempo que se tarda en capturar los datos, considerando sólo lo que se tarda en generar la estadística o ejecutar el algoritmo en cuestión.

Esta velocidad es más alta si estamos ante un sistema de analítica en tiempo real que si se trata de un sistema de analítica por lotes (del inglés batch).

Consistencia:

Se refiere en este caso a la precisión de los resultados de la analítica (no confundir, por lo tanto, con el significado de la C del teorema CAP).Tal precisión depende de si para la analítica se utilizan todos los datos disponibles (precisión alta) o de si por el contrario se emplean técnicas de muestreo para seleccionar sólo un subconjunto de los mismos con la intención de producir resultados (de menor precisión) en un menor tiempo.

Volumen:

Se refiere a la cantidad de datos que pueden ser procesados.

Hay que tener en cuenta que en entornos de Big Data, el alto volumen de datos es una características siempre presente (una de las 5 Vs).

De igual modo que hicimos al estudiar el teorema CAP, nos fijaremos en una serie de escenarios para mostrar que no podemos conseguir un sistema que cumpla a la vez las 3 características del principio SCV.

- ✔ Si se requiere velocidad (S) y consistencia (C), no podemos procesar un alto volumen (V) de datos ya que eso aumenta el tiempo de respuesta.
- ✔ Si se requiere consistencia (C) y poder procesar grande volúmenes de datos (V), no es posible realizar tal procesado a una alta velocidad (S).
- ✔ Si necesitamos procesar un alto volumen de datos (V) a una alta velocidad (S), entonces necesitaremos emplear técnicas de muestreo para seleccionar sólo un subconjunto de esos datos, lo cual producirá un resultado no consistente (C).

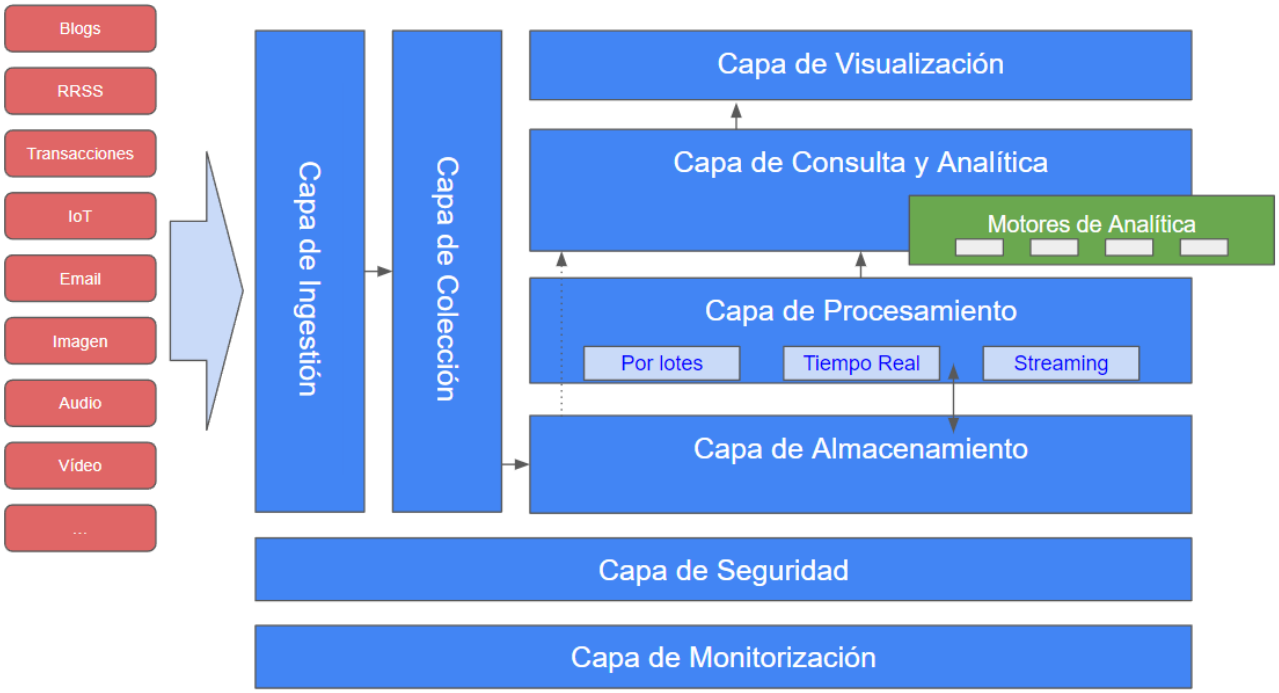
6. La arquitectura por capas de Big Data.

Al margen de que durante el diseño y desarrollo de cada posible proyecto de Big Data pueda optarse por la estructura o arquitectura que más convenga, de modo generalizado se emplea una arquitectura según la cual el flujo de datos va pasando por una serie de capas.

Capa de ingestión:
Como primer paso, los datos se obtienen desde múltiples fuentes con las cuales es necesario conectarse de algún modo. Hay que tener en cuenta que la gran mayoría de las fuentes son preexistentes a la creación del sistema Big Data que se esté desarrollando, por lo que es el sistema el que tiene que adaptarse a las fuentes y no a la inversa (empleando el protocolo correspondiente a las mismas y siendo capaz de interpetar los datos que de ellas se obtienen).

Capa de colección:
Una vez que se obtienen e interpretan los datos viene el trabajo relacionado con integrarlos para darles una estructura propia. Hay que tener en cuenta que las fuentes de datos pueden ser muchas y de naturaleza muy variada, cada una emitiendo información en un formato distinto, de modo que hay que unificarlo todo para representarlo como un único conjunto de datos con sentido y ya prácticamente listos para ser utilizados.

Capa de almacenamiento:
Como no podía ser de otro modo, esa gran cantidad de datos que da origen al concepto Big Data debe ser almacenada, para lo cual se emplean sistemas de almacenamiento distribuido especialmente diseñados para ello.



Capa de procesamiento:
Es la capa que provee de infraestructura a la siguiente capa (la de consulta y analítica)para poder tratar con gran cantidad de datos. Es decir, facilita el procesamiento (por lotes, en tiempo real, streaming o híbrido) pero únicamente hace lo que le está pidiendo la capa superior, no obteniendo valor del dato de por sí.

Capa de consulta y analítica:
Es la capa en la que se comienza a obtener valor al dato, realizando la estadística, algoritmia o análisis que se considere oportuno, para ello siempre basándose en la capa previa de procesamiento.

Capa de visualización:
Es la capa con la que interacciona el usuario final, el cual puede consultar reportes estáticos o acceder a cuadros de mando interactivos con diversas visualizaciones y controles desde los cuales puede decidir qué información ver y cómo quiere verla representada. Desde esta capa es desde por lo general se toman las decisiones de negocio.

Capa de seguridad:
Capa transversal que da soporte a todo lo relacionado con asegurar la seguridad en los datos empleando métodos tanto físicos como de software. Incluye protección ante el ataque o uso malintencionado tanto desde dentro como desde fuera de la empresa o institución.

Capa de monitorización:
Capa transversal que da soporte a todo lo relacionado con la monitorización tanto de los datos como del propio sistema. La monitorización de datos incluye auditoría, testeo, gestión y control, de modo que los datos a emplear para obtener valor sean correctos yfrescos. Tal monitorización es una parte importante de los mecanismos de gobernanza de datos.

Puedes ver más sobre lo que significa seguridad de la información en el siguiente enlace:

Seguridad de la información

Puedes ver más información sobre lo que significa gobernanza de datos en el siguiente enlace:

Gobernanza de datos

7. El Paisaje de Big Data

Hablamos de paisaje Big Data (más usado en inglés, como "The Big Data Landscape") para referirnos al panorama de las diversas herramientas y utilidades que se pueden emplear para desarrollar proyectos Big Data, muchas veces categorizadas según la capa de procesamiento a la que pertenecen o según el tipo de actividad que realizan.

Desde los inicios de Big Data, diversos autores han ido creando collages (unos más detallados que otros), tratando de capturar la riqueza de tal panorama.

Por esta razón, en esta sección no vamos a incluir una imagen en concreto sino que vamos a sugerir al alumno que realice la búsqueda "big data landscape" en su buscador preferido.

Accede a tu buscador favorito y realiza la búsqueda "big data landscape" para encontrar gran multitud de imágenes que te muestran diversas interpretaciones de lo que es el paisaje (o panorama) de Big Data.

En esas imágenes podrás encontrar, entre otras cosas:

- **Hadoop**, como la plataforma pionera para Big Data, enfocada a trabajo por lotes.
- Una gran **variedad de herramientas** pertenecientes al ecosistema de Hadoop, diseñada cada una de ellas para una función dentro de las distintas fases que componen el trabajo con datos.
- **Spark**, como plataforma enfocada a procesamiento en tiempo real y/o streaming, capaz de interactuar con muchas de las herramientas ya disponibles en el ecosistema Hadoop (de hecho según el punto de vista, Spark podría considerarse una herramienta más dentro del ecosistema).
- **Bases de datos NoSQL y NewSQL** como soluciones de almacenamiento para necesidades y casos específicos.
- Diversas **herramientas para analítica**, las cuales se pueden a su vez subclasificar.
- Diversas **herramientas para visualización**.
- Diversas **aplicaciones específicas** que se nutren de o interaccionan con alguna de las herramientas ya comentadas.