Sesión. Conceptualización y estado del arte.

Fernando López-Torrijos y José Fernando Zea

Octubre de 2021

Introducción

La analítica del Big Data es un campo orientado al análisis, procesamiento y almacenamiento de grandes colecciones de datos que, con frecuencia, provienen de diversas fuentes. Particularmente, aborda distintos requerimientos, como la combinación de múltiples conjuntos de datos no relacionados, el procesamiento de grandes cantidades de datos sin estructurar y la recopilación de información con plazos de tiempo definidos. En los entornos Big Data la analítica por lo general es aplicada usando tecnologías y plataformas distribuidas y altamente escalables. El termino de Big Data en muchos sentidos es relativo. Lo que para una entidad pueda ser Big Data, para otra no lo es. El término debe utilizarse asociado a una o más de las siguientes características: Volumen, Velocidad, Variedad, Veracidad, Volatilidad y Valor. Están interrelacionadas de muchas maneras.

1. Volumen. La cantidad de datos es tal que los sistemas de almacenamiento o de procesamiento propios no son suficientes y es necesario contratar servicios en la nube o externos. Dichos servicios son escalables horizontalmente en el sentido que se adicionan CPUs para el procesamiento o unidades de almacenamiento, las cuales procesan y/o almacenan en paralelo.

2. Velocidad. Independientemente del volumen, la tasa de generación de datos por segundo hace demasiado complejo su procesamiento. Es necesario trabajar en paralelo en un cluster de procesadores. La tira de promociones que entregan al momento de pagar las compras en el supermercado fue elaborada fracciones de segundos antes, en el momento que el cajero informa que esos son todos los artículos comprados.

3. Variedad. La fuente y la estructura sobre los cuales están contenidos es muy diversa. De hecho, contempla datos estructurados en formatos variados (XML, JSON, CSV, delimitados por tabulador, o formatos propietarios como .dat, .sav o .rds) enlazados con datos no estructurados (por ejemplo, texto escrito, video, audio y/o IoT, también en variados formatos cada uno) para enriquecerlos. Variedad es sinónimo de complejidad. Un prescriptor de rutas en el celular debe ubicar la posición del vehículo y su velocidad a partir de sensores GPS, consultar la base de datos para determinar la velocidad promedio en los últimos minutos en cada segmento de posible ruta para llegar al destino solicitado e interpretar si en la red social están informando de algún accidente o problema inusual en la ruta, ya sea mandando una foto, un audio o un mensaje del chat.

4. Veracidad. Además de contar con dimensiones de calidad tales como la exactitud del conjunto de datos, debe ser confiable también la fuente y el tipo y tratamiento de éstos, ya que estamos tratando diversidad de fuentes y estructuras.

5. Volatilidad, entendida como el ritmo de cambio y la duración de los datos. Un ejemplo de datos volátiles son los provenientes de las redes sociales, donde los sentimientos y los temas de tendencia cambian rápidamente y con frecuencia.

6. Visualización. El cerebro humano está adaptado para la visualización. La complejidad del Big Data

7. Valor. Los datos procesados, así como los resultados de los análisis, son utilizados para tareas significativas y complejas de reporte y evaluación, y además son retroalimentados en las aplicaciones para mejorar el rendimiento de estas. Interpretar los datos de la forma adecuada garantiza que los

resultados sean relevantes y generen información. El acceso a la Big Data puede implicar pasar meses clasificándola sin centrarse y sin un método claro acerca de cómo identificar qué aspectos de los datos son relevantes, pero los datos deben ser analizados en el momento oportuno, lo que resulta difícil con los Big Data, ya que de lo contrario los conocimientos no serían útiles. La tira del supermercado genera valor para el cliente si promociona artículos de su interés. La ruta recomendada por la aplicación debe contrastar la ruta recomendada con los gustos de vías del usuario según su respectivo historial y lograr que el usuario llegue en el menor tiempo posible, que es el servicio que está prestando.

Las múltiples V con la que se suele delimitar implica que es Big Data aquello que va más allá de la capacidad de la infraestructura tecnológica propia disponible, de las capacidades del personal y de los procesos usuales.

La funcionalidad del Big Data permite cuatro nuevas capacidades que ayudarán a las organizaciones a pasar de la "supervisión del quehacer" al "conocimiento del quehacer". Estos cuatro impulsores del valor del Big Data son:

1. Acceso a todos los datos transaccionales y operativos de la organización.

2. Acceso a los datos no estructurados internos y externos.

3. Explotación de la analítica en tiempo real.

4. Integración de la analítica predictiva.

Tal vez la incremental madurez de la analítica avanzada ha sido el habilitante para la generación de la funcionalidad esencial de las soluciones y herramientas de Big Data.

Ya no se trata de una moda. Se está aplicando:

El Ministerio de Hacienda cruza las páginas WEB y las ofertas en plataformas comerciales con la actividad declarada y las exenciones tributarias con el objeto de detectar elusión y lavado de activos. Básicamente textmining. (https://www.ciat.org/use-of-big-data-in-tax-administrations/?lang=en)

El Departamento Nacional de Planeación (DNP) tiene una Unidad de Científicos de Datos. Es un equipo multidisciplinario dedicado a la explotación de datos para la toma de decisiones en política pública en Colombia. Se encarga de la analítica de datos que permiten que las direcciones técnicas del DNP y otras entidades del Estado aumenten la creación de valor en sus procesos a través del aprovechamiento efectivo de la información estructurada y no estructurada, para traducirlo en una toma de decisiones objetiva. (Fuente: https://www.dnp.gov.co/programas/Desarrollo%20Digital/Paginas/Big%20Data.aspx)

MIT y *Data Pop Alliance* han estado ofreciendo apoyo técnico al Departamento Nacional de Planeación (DNP) de Colombia e iNNpulsa para diseñar, desarrollar e implementar la primera estrategia nacional de Big Data en el país. (Fuente: https://datapopalliance.org/publications/estrategia-de-big-data-para-colombia-documentos-de-diagnostico/)

La OCDE ha identificado 4 áreas de desarrollo que se ven beneficiadas en mayor medida con el uso de Big Data (Banco Mundial, 2014): i) la estimación y análisis sociodemográfico, ii) el crecimiento económico, la innovación y la investigación, iii) el análisis social, la vulnerabilidad y la resiliencia ambiental, iv) el análisis y detección de riesgos de salud pública.

Los datos de encuestas y registros administrativos se han complementado con información de telefonía móvil, redes sociales, búsquedas en Google para analizar información en tiempo real sobre lo que sucede con las personas en términos de salud, desplazamientos, y necesidades de consumo. Gran parte de los datos son generados por empresas privadas, lo que ha puesto en evidencia la necesidad de fortalecer el intercambio de datos entre el sector público y sector privado, sin dejar de proteger los derechos de los ciudadanos en materia de privacidad y seguridad de la información, así como los intereses privados de las empresas. (Fuente: https://scioteca.caf.com/handle/123456789/1776)

"La acción unilateral del sector público es inadecuada: Construcción de alianzas público-privadas" (Fuente: TASK-FORCE-IA Colombia https://dapre.presidencia.gov.co/AtencionCiudadana/Documents/TASK-

FORCE-para-desarrollo-implementacion-Colombia-propuesta-201120.pdf)

La oferta del sector privado no es poca (Fuente: https://clutch.co/co/it-services/analytics).

No obstante, todavía las compañías están en un bajo estado de madurez

(Fuente: https://www.smartdatacollective.com/five-levels-big-data-maturity-organisation/).

Las grandes consultoras tienen equipos para ayudar a madurar (Fuente: https://www.accenture.com/_acnmedia/PDF-83/Accenture-Becoming-Data-Driven-Enterprise-Data-Industrialization.pdf).

Si bien hablar de Big Data es hablar de tecnología, ésta no es un fin sino un medio. Por ello debe quedar grabada esta frase: "Las organizaciones no necesitan una estrategia de Big Data tanto como una estrategia de negocio que incorpore Big Data."

La incorporación de la Big Data dentro de las organizaciones puede realizarse para mantenerse a la par con las organizaciones del sector o para crear una diferenciación competitiva. En cualquier caso, Big Data es un medio. No un fin.

Relación entre Big data y la nube

Podemos realizar analítica sobre Big Data gracias a la computación en nube. He aquí cinco formas en las que la nube apoya a la Big Data:

1. La nube permite el acceso a la Big Data de forma rentable, de pago por uso y escalable. ¿Por qué no se guardaban y procesaban todos esos datos antes? Era demasiado costoso debido a los sistemas monolíticos que se utilizaban. Para manejar más datos, se necesitaban máquinas más grandes y el coste escalaba exponencialmente. En cambio, el Big Data basado en arquitecturas paralelas que escalan de forma lineal y elástica aprovechan los mecanismos de pago por uso y acceso bajo demanda de la nube.
2. Los servicios en la nube manejan todo lo difícil de los Big Data. Poner en marcha, gestionar y asegurar un clúster de Big Data es difícil. Los nativos de la nube lo han resuelto. No debería ser una competencia básica de todas las empresas. Los proveedores de la nube se encargan de gran parte de esta infraestructura. Gracias al poder de la nube para democratizar las TI, las empresas no tienen que abordar el Big Data como un proyecto científico arriesgado.
3. Las herramientas en la nube facilitan la experimentación con los datos. Se pueden obtener mejores conocimientos cuando es fácil trabajar con los datos. Afortunadamente, la nube ofrece herramientas para la gestión de modelos y canalizaciones de datos que permiten a los científicos e ingenieros de datos crear, experimentar y publicar modelos, conectarlos en una canalización y supervisar el rendimiento. En otras palabras, la nube se encarga de la "fontanería" de los datos para que el analista pueda centrarse en los conocimientos que pueden ayudar a su negocio.
4. La nube ayuda a gestionar los datos. Se requiere una visión unificada de los datos: quién los posee, quién puede acceder a ellos, las restricciones de privacidad, la calidad, cómo se conectan con otros datos, etc. Las herramientas emergentes de la nube ofrecen modelos de datos industriales predefinidos y sistemas de metadatos que proporcionan una visión lógica singular a través de múltiples sistemas, proveedores de la nube e incluso socios. Estos sistemas catalogan los datos de los que se dispone.
5. La nube democratiza la explotación de datos. Equipa a expertos sectoriales no técnicos que entienden la problemática de su área de experticia. Las herramientas de bajo código o gráficas, sin código, proporcionan una capacidad de *autoservicio* para que cualquier persona de la empresa pueda utilizar los datos, no sólo los expertos en datos y los ingenieros de software. El analista de negocios, el experto en la materia, el ingeniero de operaciones... todos tienen información sobre los datos al alcance de la mano gracias a la nube.

(Fuente: https://www.accenture.com/us-en/blogs/cloud-computing/cloud-and-big-data-what-you-need-to-know-in-2021-and-beyond)

Hablar de Big Data es hablar de Tecnología

Un poco de historia

El almacenamiento en la nube, del inglés cloud storage, es un modelo de almacenamiento de datos basado en redes de computadoras, ideado en los años 60 del siglo pasado, donde los datos están alojados en espacios de almacenamiento virtualizados, por lo general aportados por terceros.

Las compañías de alojamiento operan enormes centros de procesamiento de datos. Los usuarios que requieren estos servicios compran, alquilan o contratan la capacidad de almacenamiento necesaria. Los operadores de los centros de procesamiento de datos, a nivel servicio, virtualizan los recursos según los requerimientos del

cliente. Solo exhiben los entornos con los recursos requeridos. Los clientes administran el almacenamiento y el funcionamiento de los archivos, datos o aplicaciones. Los recursos pueden estar repartidos en múltiples servidores físicos.

Sólo fue hasta el ocaso del siglo pasado que se empezó a comercializar esta opción[1].

La explosión de disponibilidad de datos la proporcionó la disponibilidad de Internet y las páginas WEB.

La Web, como la conocemos hoy, se desarrolló entre marzo de 1989 y diciembre de 1990 por el inglés Tim Berners-Lee y con la ayuda del belga Robert Cailliau, mientras ambos trabajaban en el CERN en Ginebra, Suiza, y la idea fue publicada en 1991. Berners-Lee usó un NeXTcube como el primer servidor web del mundo y también escribió el primer navegador web, WorldWideWeb en 1990. El gran avance de Berners-Lee fue unir el *hipertexto* e Internet. En el proceso, desarrolló un sistema de identificadores únicos globales para los recursos web (URL) y también: el Uniform Resource Identifier.

Sin duda la posibilidad de consultar cada una de las páginas Web hizo el milagro de generar la explosión del crecimiento exponencial de la información digital. En esta disponibilidad un jugador por todos conocido es Google.

En 1996 Larry Page y Sergey Brin, siendo estudiantes, construyeron un motor de búsqueda que utilizaba enlaces para determinar la importancia de cada página en la Web. Para 1998 Andy Bechtolsheim, cofundador de Sun Microsystems, les financió constituir la empresa.

La primera oferta pública en bolsa se dio en el año 2004.

¿Qué diseñaron estos dos emprendedores que ha revolucionado el mundo?

El **Sistema de Archivos Google** (Google File System - **GFS** -), es un sistema de archivos distribuido propietario que soporta su infraestructura informática de procesamiento de información en nube. Está diseñado para proveer eficiencia, fiabilidad de acceso a datos usando sistemas masivos de clúster de procesamiento en paralelo.

En su etapa inicial se trataba de archivos divididos en porciones de tamaño fijo de 64 megabytes, similar a los clúster o sectores de las unidades de disco duro tradicional, donde muy rara vez son sobrescritos. Por lo general los archivos se adicionan o se leen. El diseño toma precauciones para prevenir un alto índice de fallas por sobrecarga en nodos individuales y, por ende, la probable pérdida de algunos datos. El diseño también apunta a manejar una gran caudal de datos y a resolver problemas de latencia.

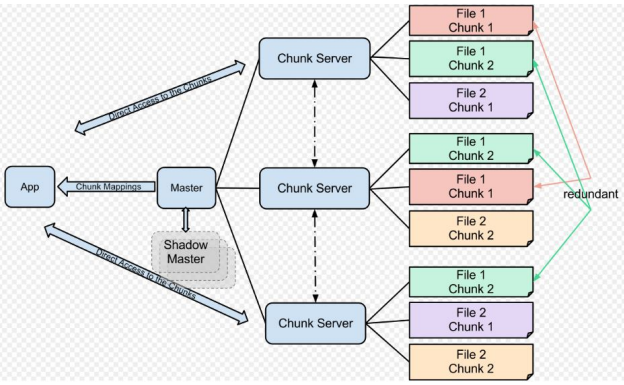

Figure 1: Esquema del Sistema de Archivos Google

El cluster GFS se compone de múltiples nodos. Estos se dividen en dos clases: un nodo Maestro y un gran número de almacenadores de fragmentos o Chunk Servers. Los archivos se dividen en porciones de tamaño fijo, los Chunk Servers almacenan las porciones, a cada una se le asigna una etiqueta de identificación única

[1]La fecha la ubican con la oferta de salesforce.com de servicios de CRM en la nube, en el año 1999. Antes, lo existente eran servicios montados por las grandes empresas para uso propio

de 64 bits en el nodo maestro al momento de ser creada, y el nodo Maestro conserva las asignaciones. A su vez cada porción es replicada en al menos tres servidores en la nube, pero también existen archivos que requieren una mayor redundancia por su enorme demanda.

Los programas acceden a las porciones mediante consultas al nodo Maestro, para localizar la ubicación de los bloques deseados. Si las porciones no se encuentran activas (por ejemplo, si no poseen accesos pendientes al almacenamiento), el nodo Maestro responde dónde están ubicadas, la aplicación contacta y recibe los datos desde el nodo de alojamiento directamente.

La principal diferencia con los demás sistemas de archivos es que el GFS no está implementado en el kernel del sistema operativo, sino que funciona como una librería en el espacio del usuario.

Para 2004 Google comenzó a desarrollar **BigTable**, un sistema de gestión de base de datos con las características de ser distribuido, de alta eficiencia y propietario. Está construido sobre el GFS, Chubby Lock Service[2] y algunos otros servicios y programas de Google, y funcionaba sobre sencillos y baratos PCs con procesadores Intel (*commodity hardware*).

BigTable almacena la información en tablas multidimensionales cuyas celdas están, en su mayoría, sin utilizar. Además, estas celdas disponen de versiones temporales de sus valores, con lo que se puede hacer un seguimiento de los valores que han tomado históricamente.

Para poder manejar la información, las tablas se dividen por columnas, y son almacenadas como 'tabletas' de unos 200 MB cada una. Cada máquina almacena 100 tabletas, mediante el sistema GFS. La disposición permite un sistema de equilibrado de carga, es decir, si una tableta está recibiendo un montón de peticiones, la máquina puede desprenderse del resto de las tabletas o trasladar la tableta en cuestión a otra máquina, y generar una rápida recomposición del sistema si una máquina *se cae*.

En ese año, 2004, Google publicó un documento en el que se describía cómo realizar operaciones en el GFS, un enfoque que pasó a conocerse como **MapReduce**. Hay dos operaciones en MapReduce: map y reduce. La operación map proporciona una forma arbitraria de transformar cada archivo en uno nuevo, mientras que la operación reduce combina dos archivos. Ambas operaciones requieren código informático personalizado, pero el marco de trabajo de MapReduce se encarga de ejecutarlas automáticamente en muchos ordenadores a la vez. Estas dos operaciones son suficientes para procesar todos los datos disponibles en la web, a la vez que proporcionan suficiente flexibilidad para extraer información significativa de ellos.

En el enlace https://burtmonroe.github.io/SoDA501/Materials/SplitApplyCombine_R/ hay un ejercicio completo de operaciones Map y Reduce con R.

Tras la publicación de estos documentos por parte de Google, un equipo de Yahoo trabajó en la implementación del Sistema de Archivos de Google y MapReduce como un proyecto de código abierto. Se lanzó en 2006 como **Hadoop**, con el Sistema de Archivos de Google implementado como Sistema de Archivos Distribuidos de Hadoop (HDFS). El proyecto Hadoop puso la computación distribuida basada en archivos al alcance de un mayor número de usuarios y organizaciones, haciendo que MapReduce fuera útil más allá del procesamiento de datos web.

Aunque Hadoop ofrecía soporte para realizar operaciones MapReduce sobre un sistema de archivos distribuido, seguía siendo necesario escribir las operaciones MapReduce con código cada vez que se ejecutaba un análisis de datos. Para mejorar este tedioso proceso, el proyecto **Hive**, lanzado en 2008 por Facebook, aportó a Hadoop soporte para el lenguaje de consulta estructurado (SQL). Esto significaba que el análisis de datos podía realizarse a gran escala sin necesidad de escribir código para cada operación de MapReduce;

[2]Chubby Lock Service es otra aplicación de Google que está pensada para su uso en un sistema distribuido de un número moderadamente grande de pequeñas máquinas conectadas por una red de alta velocidad. Por ejemplo, una instancia de Chubby (también conocida como célula Chubby) podría servir a diez mil máquinas, de 4 procesadores cada una, conectadas por 1 GB/s Ethernet. La mayoría de las células Chubby están confinadas en un único centro de datos o sala de máquinas, aunque puede haber al menos una célula Chubby cuyas réplicas están separadas por miles de kilómetros. El objetivo del servicio de bloqueo es permitir que sus clientes sincronicen sus actividades y se pongan de acuerdo sobre la información básica de su entorno. Los objetivos principales son la fiabilidad, la disponibilidad para un conjunto moderadamente grande de clientes y una semántica fácil de entender; el rendimiento y la capacidad de almacenamiento se consideraron secundarios. La interfaz de cliente de Chubby es similar a la de un simple sistema de archivos que realiza lecturas y escrituras de archivos enteros, aumentada con bloqueos de advertencia y con la notificación de varios eventos, como por ejemplo la modificación de archivos.

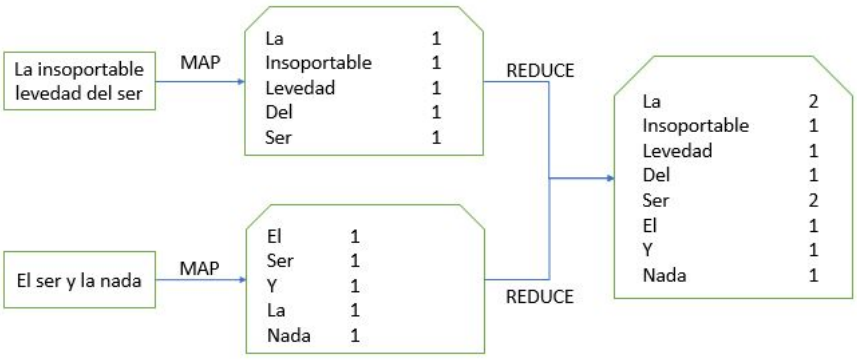

Figure 2: Ejemplo de MapReduce de conteo de palabras

en su lugar, se podían escribir sentencias genéricas de análisis de datos en SQL, que son mucho más fáciles de entender y escribir.

En 2009, Apache **Spark** comenzó como un proyecto de investigación en el AMPLab de la UC Berkeley para mejorar MapReduce. Spark ofrecía un conjunto de verbos más rico que MapReduce para facilitar la optimización del código que se ejecutaba en varias máquinas. Spark también cargaba los datos en memoria, haciendo que las operaciones fueran mucho más rápidas que el almacenamiento en disco de Hadoop.

En 2010, Spark se publicó como proyecto de código abierto y luego se donó a la Apache Software Foundation en 2013. Spark tiene una licencia Apache 2.0, que permite utilizarlo, modificarlo y distribuirlo libremente. Spark alcanzó entonces más de 1.000 colaboradores, lo que lo convierte en uno de los proyectos más activos de la Apache Software Foundation.

En 2010 se empezó a gestar **Flink**, un marco de trabajo de procesamiento de flujos y lotes unificado y de código abierto desarrollado por la Apache Software Foundation. El núcleo de Apache Flink es un motor de flujo de datos distribuido escrito en Java y Scala. Ejecuta programas de flujo de datos arbitrarios de forma paralela a los datos y canalizada, por lo tanto, paralela a las tareas. El sistema de tiempo de ejecución canalizado de Flink permite la ejecución de programas de procesamiento masivo/por lotes y de flujo. Además, el tiempo de ejecución de Flink soporta la ejecución de algoritmos iterativos de forma nativa.

Flink proporciona un motor de streaming de alto rendimiento y baja latencia, así como soporte para el procesamiento de eventos y la gestión de estados. Las aplicaciones de Flink son tolerantes a fallos en caso de que la máquina falle y soportan la semántica de exactamente una vez[3]. Los programas pueden escribirse en Java, Scala, Python, y SQL y se compilan y optimizan automáticamente en programas de flujo de datos que se ejecutan en un entorno de clúster o nube.

Flink no proporciona su propio sistema de almacenamiento de datos, pero proporciona conectores de intermediación de mensajes a sistemas como Amazon Kinesis, Apache Kafka, HDFS, Apache Cassandra y ElasticSearch.

Por ejemplo, Apache Kafka es un proyecto de intermediación de mensajes de código abierto desarrollado por LinkedIn y donado a la Apache Software Foundation en 2012. El proyecto tiene como objetivo proporcionar una plataforma unificada, de alto rendimiento y de baja latencia para la manipulación en tiempo real de fuentes de datos. Puede verse como una cola de mensajes, bajo el patrón publicación-suscripción, masivamente escalable concebida como un registro de transacciones distribuidas, lo que la vuelve atractiva para las infraestructuras de aplicaciones empresariales.

En informática, **NoSQL** es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de sistemas de gestión de bases de datos relacionales en aspectos importantes, siendo los más

[3]Los registros no se pierden nunca, pero pueden volver a entregarse. Si la aplicación de procesamiento de flujos falla, no se pierde ningún registro de datos y no se procesa, pero algunos registros de datos pueden volver a leerse y, por lo tanto, volver a procesarse.

destacados que no garantizan atomicidad, consistencia, aislamiento y durabilidad (ACID) y que habitualmente escalan bien horizontalmente. Los datos almacenados no requieren estructuras fijas como tablas y normalmente no soportan operaciones JOIN. Los sistemas NoSQL se denominan a veces "no solo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL. Las empresas de Facebook utilizan este tipo de esquema.

Cassandra y MongoDB son dos tecnologías de almacenamiento NoSQL muy difundidas actualmente.

Cassandra es una tecnología de código abierto creada en 2008 por Facebook que combina las tecnologías de sistemas distribuidos del almacén de valores clave de Amazon Dynamo y el modelo de datos basado en columnas BigTable de Google. Es una solución escalable con una *familia de columnas* que proporciona a los usuarios la capacidad de almacenar grandes cantidades de datos estructurados y no estructurados. Tiene la capacidad de escalar a través de múltiples centros de datos lo cual proporciona *disponibilidad* frente a *interrupciones* o *caídas*.

Las tablas de Cassandra utilizan un *esquema*, como una base de datos relacional tradicional, pero elimina la noción de *normalización* de la base de datos estructurada. La razón es que el almacenamiento en una red de servidores básicos es barato, por lo que no hay razón para hacer uniones para recuperar datos, lo que ralentiza la recuperación de datos. En su lugar, la redundancia se considera adecuada.

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas las guarda en estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Tiene drivers para múltiples lenguajes de programación.

Un registro de datos JSON es autodescriptivo, porque el nombre del campo y el valor de los datos se almacenan en el mismo lugar, es decir, dentro del documento. Aunque no se requiere un *esquema* con MongoDB, ya que JSON por definición no necesita uno, se puede hacer uno.

Tendencias emergentes en el Big Data

1. Mayor dependencia del almacenamiento en la nube.

Con el crecimiento de la tecnología, con la recopilación de datos en tiempo real o flujos de datos y el conocimiento ganado acerca de cómo se pueden utilizar estratégicamente los datos, la capacidad de almacenamiento de Big Data es un problema.

En la mayoría de las empresas, el almacenamiento tradicional de datos en las instalaciones ya no es suficiente para los terabytes y petabytes de datos que fluyen en la organización. Cada vez se eligen más las soluciones en la nube y en la nube híbrida por su infraestructura de almacenamiento simplificada y su escalabilidad. Y con el aumento de la dependencia del almacenamiento en la nube, también se ha empezado a implantar otras soluciones basadas en la nube, como los almacenes de datos alojados en la nube y los lagos de datos.

Por ejemplo, **Snowflake** es una compañía de almacenamiento de datos basada en computación en la nube. Fue fundada en 2012 y lanzada al público en octubre de 2014. Ofrece almacenamiento de datos en la nube y servicios analíticos, generalmente denominados "almacenamiento de datos como servicio". Permite a los usuarios corporativos almacenar y analizar datos usando hardware y software basado en la nube. Corre sobre **Amazon S3** desde 2014, sobre **Microsoft Azure** desde 2018 y en la Plataforma **Google cloud** desde 2019.

2. El crecimiento de la tecnología del *Tejido de datos*.

Hace referencia a poder almacenar y recuperar fácilmente los conjuntos de datos a través de la infraestructura distribuida en las instalaciones, la nube y la red híbrida.

Un **servidor híbrido** es un nuevo tipo de servidor virtual que ofrece tanto la potencia de un servidor dedicado clásico como la flexibilidad del cloud computing. En los servidores híbridos el hardware se comparte entre los usuarios. El precio es inferior al de los servidores dedicados. Se ofrece a un coste fijo que se basa en varios mecanismos de financiación, como el gasto de capital con depreciación lineal desde el precio de compra hasta el valor residual y la opción de arrendamiento o financiación.

El servidor se separa en entornos de servidores híbridos utilizando cualquier sistema de virtualización. Cada entorno híbrido está aislado de forma segura y dispone de recursos garantizados que aseguran un alto nivel de rendimiento y capacidad de respuesta. Un servidor híbrido combina las ventajas de la tecnología de virtualización con el rendimiento de un servidor dedicado. Así, un administrador puede utilizar la automatización para suspender, reiniciar o reinstalar el sistema operativo. Un servidor grande se divide en varios (normalmente dos) servidores híbridos. Las ventajas de esta plataforma también incluyen el acceso a través de un único punto de contacto, el uso compartido de la infraestructura de red y la supervisión, entrega y gestión de los servicios de alojamiento.

Esta tecnología se está desarrollando progresivamente en la nube y están siendo adoptada por las organizaciones que necesitan un espacio adicional y una mayor accesibilidad para sus crecientes grupos de Big Data.

La tecnología de tejido de datos también es tendencia en el mundo de la inteligencia artificial (IA) y la automatización del aprendizaje automático (ML) para el Big Data principalmente porque el diseño distribuido desalienta los silos de datos que dificultan la anotación de datos[4] y el aprendizaje automático.

3. Recopilación ética de datos.

Gran parte del aumento de los Big Data a lo largo de los años ha llegado en forma de datos de los consumidores o datos que están constantemente conectados a los consumidores mientras usan tecnologías como dispositivos de streaming, dispositivos IoT y redes sociales. El cumplimiento de las regulaciones internacionales como la *General Data Protection Regulation* de la Unión Europea se vuelve complicado cuando las empresas no saben de dónde provienen sus datos o qué datos sensibles se almacenan en sus sistemas. Se resuelve aplicando una recopilación ética de datos de los clientes.

Se vuelve más escasa la información disponible para compra en el mercado, por tanto las empresas recopilan ellas mismas los daros no sólo para garantizar el cumplimiento de las leyes sino para mantener la calidad de los datos y para ahorrar costos.

4. Automatización impulsada por la IA y/o ML.

Una de las mayores tendencias del Big Data es el uso la Inteligencia Artificial y el Machine Learning para automatizar el manejo del Big Data, tanto para las necesidades de los consumidores como para las operaciones internas de las organizaciones.

Por supuesto, este tema levanta interrogantes éticos.

5. Búsqueda de similitudes vectoriales.

La búsqueda de *similitudes vectoriales* es un nuevo método de búsqueda a través de Big Data que utiliza una combinación de modelos de aprendizaje profundo y algoritmos de última generación para encontrar elementos por sus significados conceptuales en lugar de por palabras clave o propiedades.

Fuente: https://www.datamation.com/featured/big-data-trends/

Retos persistentes del Big data.

Tipos de retos:

- Integrar grandes y complejos conjuntos de datos.
- Empezar con el proyecto de Big Data correcto.
- Desarrollar e implementar la infraestructura para administrar y procesar el Big Data
- La carencia de talento humano o asesores con la pericia para darle sentido a la Big Data.
- Un reto es tratar con datos inexactos o ambiguos.

[4]La anotación de datos es la categorización y el etiquetado de datos para aplicaciones de IA. Los datos de entrenamiento deben organizarse y anotarse adecuadamente para un caso de uso específico. Con una anotación de datos de alta calidad y realizada por personas, las empresas pueden crear y mejorar las aplicaciones de IA.

¿Big data o no Big data?

No siempre los proyectos de Big Data son los más adecuados, cuando no se disponen de los recursos ni conocimientos de las tecnologías, el muestreo probabilístico es una gran alternativa, especialmente para almacenar registros administrativos. Por otro lado, no siempre los registros administrativos o información de una empresa están diseñados para propósitos estadísticos, se requiere un gran esfuerzo de limpieza de datos, validación y disposición en formatos adecuados para que los registros administrativos pueden responder a los requerimientos de una organización. Hay información no estructurada de gran valor para las organizaciones, imágenes, video o audio que no se explota para la estrategia de las organizaciones. En este caso las estrategias de muestreo no son tan claras. El problema de sesgo acompaña constantemente a la información de las organizaciones, las metodologías estadísticas y conocimientos de inferencia causal siguen estando a la orden del día: **Everybody Lies: Big Data, New Data, and What the Internet Can Tell Us About Who We** (https://www.amazon.com/-/es/dp/B06XCYD5KG/ref=sr_1_13?___mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=big+data&qid=1633671971&sr=8-13)

Información estructurada y no estructurada

Los datos de una organización pueden dividirse en datos estructurados (los datos usuales dispuestos en tablas en filas y columnas), datos semiestructurados (archivos json, y xml) comunes en el desarrollo de aplicaciones web para intercambiar información y usado ampliamente en NoSQL y los datos no estructurados, por ejemplo, imágenes, video, audio y texto las cuales requieren un tratamiento especializado para poder aprovechar su información.

Las imágenes constituyen valiosas fuentes de información, por ejemplo, las imágenes de productos incluidas en catálogos o publicidad en espacios televisivos constituye una gran fuente de información. Las imágenes satelitales pero comprender los cultivos desarrollados en un país o las imágenes que permitan seguir cumplimiento de metas de reforestación constituyen información no estructurada que puede convertirse en información valiosa.

La información estructurada y no estructurada pueden obtenerse de información de organizaciones, de sistemas CORE, de Web-scrapping de repositorios de datos, imágenes y audios, de información transmitida por dispositivos electrónicos (Internet de las cosas).

A continuación, se ilustra cómo se "estructura" la información dispersa en las imágenes:

Figure 3: Imagen de ejemplo

Las coordenadas de los pixeles se ubican como se describe la imagen:

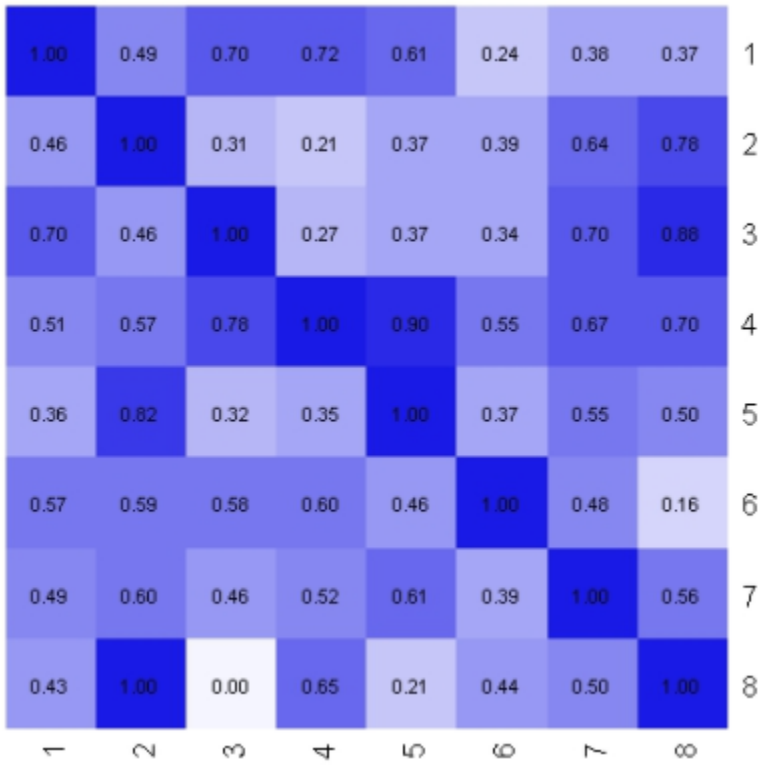

Figure 4: Coordenadas de una imagen

La imagen anterior está constituida por 50.325 pixeles y viene en tres gamas de colores, algunos de estos colores son:

- Blanco: 1, 1, 1
- Negro: 0, 0, 0
- Rojo: 1, 0, 0
- Blue: 0, 1, 0
- Verde: 0, 0, 1

Una página de internet muy interesante para revisar los colores es: https://convertingcolors.com, es de destacar que los colores Rojo (R), Azul (B) y Verde (G) no toman valores entre 0 y 1 sino entre 0 y 255. Las imágenes están dispuestas en arreglos tridimensionales, se pueden pensar como tres matrices con una distribución de pixeles en 183 recuadros a lo ancho y 275 recuadros a lo largo.

```
## [1] 183 275   3
```

```
## [1] 183 275
```

Revisaremos los colores, observe que el color de la esquina superior izquierda es blanco:

```
## [1] 1
```

```
## [1] 1
```

```
## [1] 0.9921569
```

El color de la esquina superior derecha es un tono cercano al verde y al azul (ver la página)

```
## [1] 0.2039216
```

```
## [1] 0.3529412
```

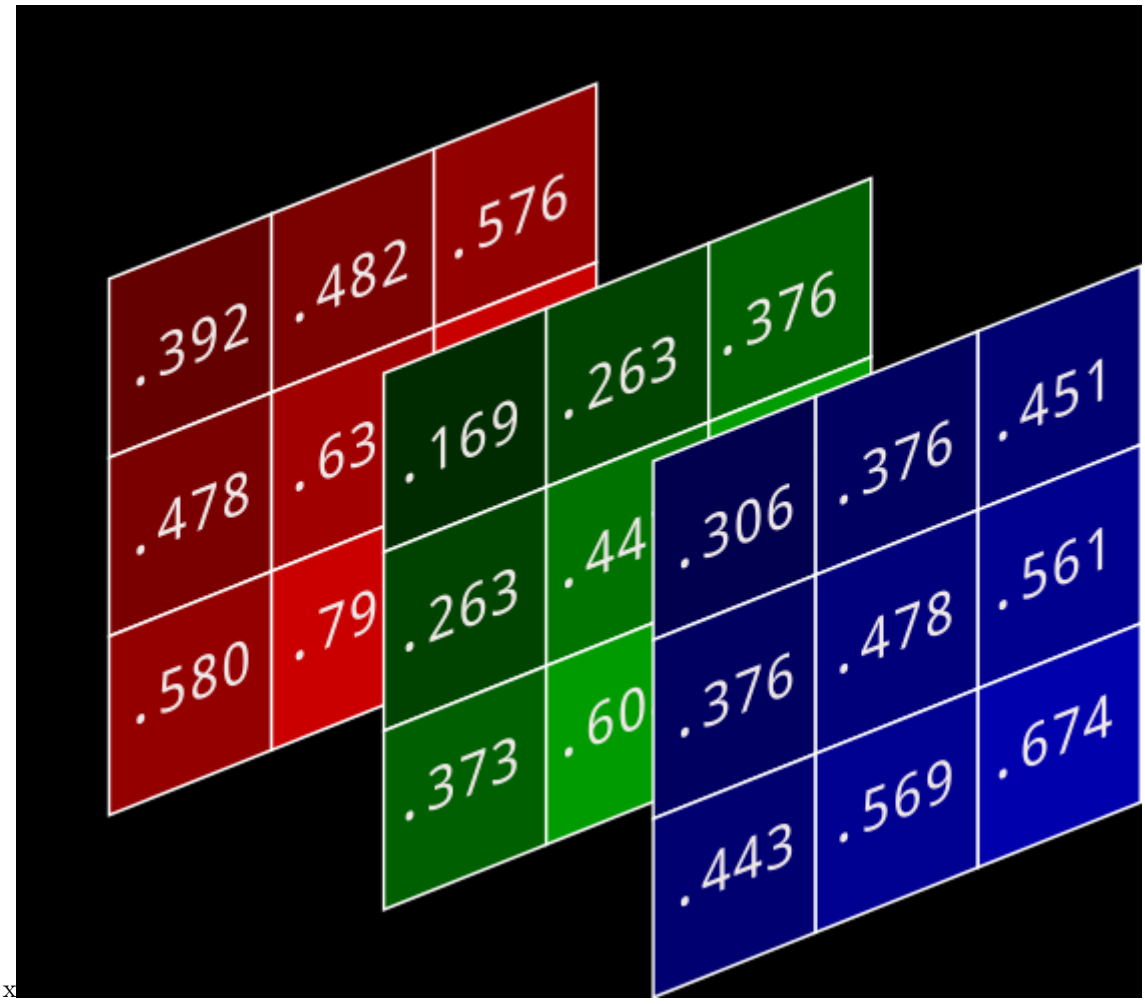

Figure 5: Coordenadas de una imagen

```
## [1] 0.4941176
```

En la esquina inferior izquierda se observa verde:

```
## [1] 0.4431373
```

```
## [1] 0.4431373
```

```
## [1] 0.09803922
```

En la esquina inferior derecha se observa un color café:

```
## [1] 0.8823529
```

```
## [1] 0.5176471
```

```
## [1] 0.3882353
```

El color del centro es un tono rosado:

```
## [1] 1
```

```
## [1] 0.4470588
```

```
## [1] 0.4705882
```

Con la información dispuesta en imágenes se pueden resolver problemas de clasificación, reducción de dimensionalidad entre muchos otros.

Otro tipo de información no estructurada que comúnmente se encuentra en la práctica es información textual, a continuación, se leen 13 documentos del congreso de los diputados de España en documentos pdfs:

Se realiza la lectura de manera que se detecte las dos columnas en el texto, en pdftools las dos columnas no se detectan correctamente, si con el paquete tabulizar, inmediatamente se corrige las palabras separadas por guiones, se eliminan los saltos de página, las puntuaciones y espacios en blanco y se convierten los textos a minúsculas, además se coloca en un formato "tidy" los textos.

```
##           used (Mb) gc trigger (Mb) max used (Mb)
## Ncells  912401 48.8    1799581 96.2   912401 48.8
## Vcells 1739395 13.3    8388608 64.0  1739395 13.3
```

```
## 26.54 sec elapsed
```

A continuación, se ilustra una manera de disponer la información proveniente de documentos, la matriz de frecuencia de documentos (filas) y términos, conocida en inglés como Document Term Matrix:

```
## <<DocumentTermMatrix (documents: 13, terms: 10)>>
## Non-/sparse entries: 40/90
## Sparsity           : 69%
## Maximal term length: 11
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs abandonarán abandono abarca abastecer abastecerse abierta abismales
##   1            1        1      1         1           1       1         1
##   10           0        0      0         0           0       1         0
##   11           0        1      0         0           0       3         0
##   12           0        6      0         0           0       2         0
##   13           0        0      2         0           0       0         0
##   3            0        0      0         0           0       0         0
##   4            0        0      1         0           0       1         0
##   6            0        0      0         0           0       1         0
##   8            0        3      1         0           0       2         0
```

```
##   9           0        0      1         0           0       0         0
##     Terms
## Docs abocados aborde abre
##   1         3      1    2
##   10        0      0    3
##   11        0      0    5
##   12        1      1    2
##   13        0      0    2
##   3         0      0    2
##   4         2      0    0
##   6         1      1    2
##   8         0      0    2
##   9         0      0    3
```

Con estas matrices se generan análisis estadísticos, modelos de machine learning y se realiza en general analítica avanzada. Es frecuente trabajar con un formato conocido como tdf-idf para llevar a cabo procedimientos estadísticos con la información:

```
## Warning in TermDocumentMatrix.SimpleCorpus(x, control): custom functions are
## ignored
```

```
## <<DocumentTermMatrix (documents: 13, terms: 5)>>
## Non-/sparse entries: 12/53
## Sparsity           : 82%
## Maximal term length: 11
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
## Sample             :
##     Terms
## Docs  abandonarán     abandono       abarca    abastecer  abastecerse
##   1  0.0001658795 7.622556e-05 6.179450e-05 0.0001658795 0.0001658795
##   11 0.0000000000 4.867857e-05 0.000000e+00 0.0000000000 0.0000000000
##   12 0.0000000000 1.387340e-04 0.000000e+00 0.0000000000 0.0000000000
##   13 0.0000000000 0.000000e+00 1.974238e-04 0.0000000000 0.0000000000
##   2  0.0000000000 0.000000e+00 0.000000e+00 0.0000000000 0.0000000000
##   3  0.0000000000 0.000000e+00 0.000000e+00 0.0000000000 0.0000000000
##   4  0.0000000000 0.000000e+00 3.903142e-05 0.0000000000 0.0000000000
##   5  0.0000000000 0.000000e+00 0.000000e+00 0.0000000000 0.0000000000
##   8  0.0000000000 1.734553e-04 4.687221e-05 0.0000000000 0.0000000000
##   9  0.0000000000 0.000000e+00 1.162222e-04 0.0000000000 0.0000000000
```

Ejercicio Ilustrar como se estructura la información proveniente de audio.