Big data, machine learning y data science en python

Big data, machine learning y data science en python

José Manuel Ortega

Ra-Ma®

La ley prohíbe
fotocopiar este libro

Big data, machine learning y data science en python
Materia: GPH - Ciencia y análisis de datos

Editado por:
RA-MA Editorial
Calle Jarama, 3A, Polígono Industrial Igarsa
28860 PARACUELLOS DE JARAMA, Madrid
Teléfono: 91 658 42 80
Fax: 91 662 81 39
Correo electrónico: *editorial@ra-ma.com*
Internet: *www.ra-ma.es* y *www.ra-ma.com*
ISBN impreso: 978-84-1944-458-5
ISBN ePub: 978-84-19444-59-2
Depósito legal: M-319-2023
Maquetación: Antonio García Tomé
Diseño de portada: Antonio García Tomé
Filmación e impresión: Safekat
Impreso en España en enero de 2023

A mi familia.

ÍNDICE

OBJETIVOS

El libro está dirigido aquellos lectores que estén trabajando en proyecto relacionados con big data y busquen identificar las características de una solución de Big Data, los datos asociados a estas soluciones, la infraestructura requerida, y las técnicas de procesamiento de esos datos. Entre los principales **objetivos** podemos destacar:

- Introducir los conceptos de ciencias de datos y machine learning.
- Introducir las principales librerías que podemos encontrar en Python para aplicar técnicas de machine learning a los datos.
- Dar a conocer los pasos para construir un modelo de machine learning, desde la adquisición de datos, pasando por la generación de funciones, hasta la selección de modelos.
- Dar a conocer los principales algoritmos para resolver problemas de machine learning.
- Introducir scikit-learn como herramienta para resolver problemas de machine learning.
- Introducir pyspark como herramienta para aplicar técnicas de big data y map-reduce.
- Introducir los sistemas de recomendación basados en contenidos.

El libro trata de seguir un enfoque teórico-práctico con el objetivo de afianzar los conocimientos mediante la creación y ejecución de scripts desde la consola de Python. Además, se provee un repositorio donde se pueden encontrar los ejemplos que se analizan a lo largo del libro para facilitar al lector las pruebas y asimilación de los contenidos teóricos.

1

INTRODUCCIÓN A BIG DATA

1.1 INTRODUCCIÓN

En el presente capítulo se va a detallar las diferentes arquitecturas utilizadas en un ecosistema Big Data y las capas más importantes como seguridad, gestión, generación, adquisición, almacenamiento y análisis, los actores, tecnologías y herramientas que forman parte de la arquitectura. Estas tecnologías y herramientas son implementadas según las características del proyecto o tipo de investigación a realizar, es por eso que se van a definir los componentes funcionales de una arquitectura Big Data, resaltando en qué casos suelen ser más útiles o cómo en combinación con otras pueden aportar mejores resultados.

Si hablamos de Big Data, esta no es una sola tecnología, sino una combinación de viejas y nuevas tecnologías que se integran para poder abordar las nuevas características de los datos como velocidad, variedad y volumen. Por lo tanto, Big Data es la capacidad de manejar un gran volumen de datos de diversas fuentes, a la velocidad correcta, y dentro del marco de tiempo adecuado para permitir el análisis ya sea posterior a la recolección de los datos o en tiempo real. Big Data está típicamente dividido en tres características que son las 5Vs.

El volumen que es la cantidad de datos, la velocidad que hace referencia la tasa de flujo de los datos en la creación, almacenamiento, análisis y visualización, y variedad que son las distintas fuentes de datos. Aunque se tiende a simplificar Big Data en 5Vs existen propuestas que hacen referencia a otras como la variabilidad que se refiere a cualquier cambio de los datos en el tiempo como puede ser la tasa de transferencia o el formato, la veracidad la cual indica la exactitud o precisión de los datos.

Por lo que no debe entenderse la definición de Big Data limitada a solo 5Vs; por ejemplo, puede darse el caso de una cantidad relativamente pequeña de datos muy diversos y complejos o es posible que se procese un gran volumen de datos muy simples. Esos datos simples pueden ser estructurados, semiestructurados o no estructurados. Es por eso que se suele incluir la V de valor que hace referencia al aporte de valor a la organización de parte del análisis de los datos a través del procesamiento Big Data. Esto nos indica, por ejemplo, cuán precisos son los datos elegidos para predecir el valor del negocio o si en realidad tiene sentido los resultados del análisis de Big Data.

1.2 DEFINICIÓN DE BIG DATA

Big Data o datos a gran escala hace referencia a un conjunto de datos tan grande que las aplicaciones informáticas tradicionales de procesamiento de datos no son capaces de tratar con ellos ni de encontrar patrones repetitivos. Se encuentra dentro del sector de las tecnologías de la información y la comunicación (TIC) y se ocupa de la manipulación y procesamiento de grandes volúmenes de datos.

Big Data es la agrupación de múltiples tendencias tecnológicas, maduradas a partir del año 2000. Dichas tecnologías se han consolidado entre los últimos años, momento en el que la sociedad se encuentra generando información alrededor de las redes sociales, un mayor ancho de banda, reducción de los costes de conexión a internet, telefonía móvil, internet de las cosas y computación en la nube.

La popularización de Big Data ha venido explicada inicialmente por 3 Vs: el procesamiento de grandes **volúmenes** de datos que llegan a grandes **velocidades** y con una **variedad** de fuentes de información nunca vista hasta ahora. En el modelo en V de Big Data se proponen 5 grupos de procesos:

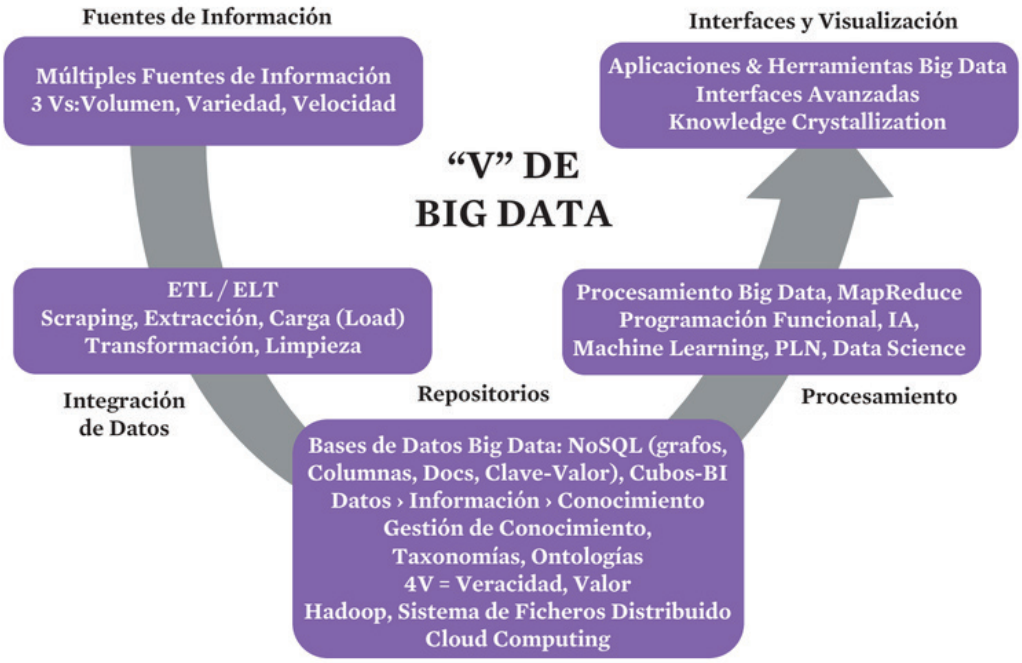

Figura 1.1. Modelo de proceso en Big Data

- **Fuentes de Información Big Data:** enriquecemos nuestras fuentes de datos con nuevas fuentes disponibles de forma abierta en internet. Toda esta variedad de fuentes de información genera grandes volúmenes de datos que llegan a gran velocidad. Las taxonomías que clasifican esas fuentes son relevantes.
- **Integración de datos Big Data**: extraemos los datos y los cargamos en Repositorios de Información especialmente diseñados para tratar Big Data. Frente a la posibilidad de transformar y limpiar los datos antes de cargarlos la tendencia es cargar todos los datos para poder explotarlos a posteriori para otros fines. Cobra asimismo importancia el proceso de Scraping de información, de lectura de datos directamente de la web mediante aplicaciones software que llamamos Bots.
- **Sistema y Repositorios Big Data**: nuevos tipos de Bases de Datos, que llamamos NoSQL son los nuevos contenedores de información, especialmente preparados para los tipos de procesamiento necesarios. Además de datos e información gestionamos el

conocimiento en Ontologías, que son reflejo de una 4a V, la Veracidad. El Sistema de Ficheros Distribuido y el Cloud Computing son la base de este Sistema Big Data.

- **Procesamiento Big Data:** tecnologías tradicionales como la programación funcional, el machine learning, el procesamiento de lenguaje natural, y un grupo de áreas de conocimiento que agrupamos bajo los paraguas de la "Data Science" y la Inteligencia Artificial se aprovechan de nuevas capacidades de procesamiento distribuido y masivo de datos para ser el 4o eslabón de la "V" de Big Data. En torno a este grupo de procesos aparece para algunas empresas una 5a "V", la Viscosidad, referenciando con ese concepto la mayor o menor facilidad para correlacionar los datos.
- **Interfaces y Visualización Big Data:** los usuarios necesitan nuevos sistemas de visualización, interacción y análisis para interactuar con el Big Data, diferentes a los tradicionales provenientes del mundo del Business Intelligence. Aparecen situaciones en las que, por ejemplo, una misma pregunta cristaliza en diferentes respuestas para diferentes usuarios según su contexto.

La consultora Gartner lo describe como "Big Data son los grandes conjuntos de datos que tiene tres características principales: **volumen** (cantidad), **velocidad** (velocidad de creación y utilización) y **variedad** (tipos de fuentes de datos no estructurados, tales como interacción social, video, audio, cualquier cosa que se pueda clasificar en una base de datos)".

El ritmo actual de generación de datos está sobrepasando las capacidades de procesamiento de los sistemas actuales en compañías y organismos públicos. Las redes sociales, el Internet de las Cosas y la industria 4.0 son algunos de los nuevos escenarios con presencia de datos masivos.

La necesidad de procesar y extraer conocimiento valioso de tal inmensidad de datos se ha convertido en un desafío considerable para científicos de datos y expertos en la materia. El valor del conocimiento extraído es uno de los aspectos esenciales del Big Data.

Con el objetivo de cubrir la problemática existente del almacenamiento, tratamiento y aprovechamiento de los grandes volúmenes de datos que se producen en la actualidad por factores como son: la elevada y creciente cantidad de fuentes de datos (sensores y redes sociales, por ejemplo) y la generalización de las redes de telecomunicaciones, en muchos casos inalámbricas. El conjunto de estos elementos, junto con las mayores capacidades de almacenamiento, ha hecho crecer de una manera enorme la cantidad de datos disponibles en los últimos años, tendencia que se sigue manteniendo en la actualidad.

Otra posible definición es la que describe Big Data a través de tres características:

- **Volumen**: gran cantidad de datos.
- **Velocidad**: procesamiento cercano a tiempo real.
- **Variedad**: distintas fuentes de información y formato.

La primera de las características más importantes de este concepto hace referencia a la circunstancia de que la cantidad de datos que se manejan supera actualmente el desproporcionado rango de los Exabytes de información. Obviamente, toda esta gran cantidad de datos puede obtenerse de diversas fuentes o ser presentados en infinidad de formas (variedad).

El volumen se incrementa en órdenes de magnitud no vistos anteriormente en los almacenes de información tradicionales, hablándose incluso de Zetabytes. Por otra parte los datos empiezan a llegar a los sistemas en tiempo real (Velocity) y hay que ser capaz de tratar esa información para que no se pierda nada.

Por último, empiezan a llegar fuentes de datos eminentemente desestructuradas (básicamente texto procedente de Internet) que siguen conviviendo con las fuentes estructuradas clásicas, aquí

estamos hablando de variedad (Variety) en las fuentes de información que será necesario integrar para tener una visión global de cada escenario.

Todas las aplicaciones que hacen uso de estos datos necesitan obtener unos tiempos de respuesta mínimos que permitan lograr la obtención de la información correcta en el momento preciso. Esta información debe ser lo más veraz posible; es decir, las fuentes de las cuáles se obtiene deben ser lo más fiable posible para así poder generar el valor tan ansiado que haga que nuestros datos sirvan para un fin concreto, como puede ser la toma de decisiones críticas en organizaciones o la comprobación de la evolución del tráfico en un portal de Internet, por ejemplo.

Debido a esto, en el mundo en el que nos encontramos es necesario determinar qué información queremos obtener, para que el volumen de los datos no nos desborde. Para tal fin, se utilizarán un conjunto de herramientas que permitan el almacenamiento, procesamiento, recuperación y análisis de una cantidad inmensa de datos.

Big Data se suele definir como "conjunto de técnicas que permiten analizar, procesar y gestionar conjuntos de datos extremadamente grandes que pueden ser analizados informáticamente para revelar patrones, tendencias y asociaciones". Además, el volumen no tiene definido un tamaño mínimo que divida, lo que es Big Data y lo que no. Según un estudio, no existe una cantidad de datos específica, aunque afirma que usualmente se habla en términos de petabytes y exabytes de datos.

- **Gigabyte**: equivale aproximadamente a 256 canciones si el tamaño promedio de cada canción son 4 MB.
- **Terabyte**: : cantidad equivalente a 4 portátiles de 256 GB, teniendo en cuenta que el S.O. ocupa parte de ese espacio.
- **Petabyte**: todas las fotos que posee Facebook equivalen a 1.5 PB.
- **Exabyte**: Empresas como Google, Amazon o Facebook suelen manejar tales cantidades de datos.

La capacidad de cómputo del hardware y el software crece exponencialmente. Hoy en día tenemos en nuestro bolsillo, concretamente en nuestros modernos teléfonos móviles, más capacidad de cómputo que los ordenadores de la NASA que llevaron al hombre a la luna. Los ordenadores personales de los que disponíamos a finales de los años 90 son hoy tristes antiguallas, apenas útiles más que en exposiciones de juegos retro.

En los últimos años han evolucionado tanto las técnicas como las nuevas capacidades del hardware y del software que nos hacen posible usar ahora paradigmas informáticos de altas capacidades que hasta hace pocos años eran computacionalmente inviables.

Estas nuevas tecnologías pueden habilitar nuevas capacidades para las organizaciones fundamentadas en el término paraguas Big Data, materializadas en servicios, funciones u operaciones nuevas o muy mejoradas. La implementación de estas nuevas capacidades puede conseguir como resultado importantes beneficios.

Big Data como paradigma también nos ha aportado Sistemas de Archivos Distribuidos y escalables y nuevos sistemas de gestión de bases de datos preparados para dar respuesta a la necesidad de manejar grandes volúmenes de información de forma distribuida. Ejemplos hoy de rabiosa actualidad son las **Bases de Datos NoSQL**, entre las que destacan las orientadas a columnas, las de clave-valor, las orientadas a la gestión de documentos, objetos o grafos.

Los otros enfoques emergentes son los del Aprendizaje Automático, popularmente conocido por su denominación en inglés, "**Machine Learning**", y los Métodos Probabilísticos

y Estadísticos. Estos dos enfoques, aplicados tanto a textos desestructurados como a datos masivos, proporcionan resultados novedosos aplicados a los procesos analíticos, prospectivos y predictivos.

En **Machine Learning** utilizamos conjuntos de información y un algoritmo para entrenar a una aplicación. Una vez entrenada, cada vez que necesitemos analizar una nueva información dicha aplicación clasificará la nueva información a partir del entrenamiento recibido. En el algoritmo de entrenamiento podemos estar utilizando tanto los métodos probabilísticos y estadísticos mencionados anteriormente como otras técnicas de inteligencia artificial como redes neuronales, árboles de decisión, etc.

Los métodos probabilísticos y estadísticos nos van a ofrecer un modelo de referencia para un conjunto de datos, gracias al cual podamos clasificar una nueva información ofreciendo una predicción a partir de dicho modelo. Estos modelos se aplican tanto a datos numéricos como a conjuntos de palabras dentro de documentos. Son aplicados actualmente, por ejemplo, por los grandes buscadores de Internet para determinar qué documentos son más relevantes para una búsqueda dada.

Para agrupar todo este conocimiento que se está concentrando en torno al término de Big Data ha emergido el concepto de Data Science. Las implementaciones Big Data serían imposibles sin las nuevas capacidades de los ordenadores actuales, que han evolucionado enormemente tanto en el hardware como en el software. Además de la capacidad de procesamiento, el almacenamiento es el otro punto en el que el hardware ha evolucionado: el coste de un dispositivo de 1Gb de capacidad ha disminuido de 300.000 € en 1980, a unos 10 € en el año 2000 y apenas unos céntimos en la actualidad.

En cuanto al software las claves están en la evolución y mejora de los sistemas operativos y en la virtualización, encarnada en las máquinas virtuales, un software capaz de emular a una computadora, pudiendo ejecutarse en un mismo ordenador varias máquinas virtuales. Ambas evoluciones, de hardware y software, han habilitado una paralelización potente y fiable, haciendo posible poner a funcionar en paralelo cientos o miles de estos ordenadores que, aplicando el viejo lema de Julio César "divide et vinces", divide y vencerás, separamos los problemas en multitud de pequeños problemas fáciles de solucionar y luego integran todas esas pequeñas soluciones en la solución final del problema planteado, todo ello realizado en un intervalo de tiempo pequeño. A este tipo de sistemas lo llamamos **sistemas distribuidos**.

Gracias a todo esto se ha habilitado la posibilidad de que en grandes centros de datos se implementen todas estas nuevas capacidades de cómputo y se le ofrezcan nuevos servicios al mercado. A este otro paradigma lo llamamos "**Cloud Computing**", computación remota, en definitiva.

Por último, la aparición de proyectos de software libre, entre los que destaca el **Apache Hadoop**, ha hecho posible esta revolución. Las grandes empresas de internet han promovido un uso masivo de software libre principalmente por su capacidad de adaptación rápida a sus nuevas necesidades, pero también hay que mencionar que el reducido o inexistente coste de licencias del mismo ha posibilitado la viabilidad económica de estas empresas.

Big Data contempla las nuevas herramientas, tecnologías y (nuevos) los conceptos relacionados con la adquisición de (mucha) data (volumen), de distinto tipo (variedad) que a su vez podría estar no estructurada, con unos aspectos opcionales pero que también puede marcar la diferencia para definirlo como "really Big Data" como la movilidad (por ejemplo la adquisición de información mediante IoT o dispositivos móviles) y el tiempo real. De hecho al trabajar con Big Data se podrían considerar las siguientes vertientes que pueden o no trabajar en conjunto:

- **Ingeniería:** Enfocado en el uso de las herramientas por ejemplo al tratar verdaderamente mucha data con poco o nada de análisis, un rol de esta vertiente sería el Arquitecto de Datos, esa persona encargada de manipular estructurar los datos, manipularlos, masticarlos y dejarlos bien preparados para aquellos encargados de hacer análisis sobre los datos, esta persona trabajaría con Hadoop, Pig, Spark.
- **Científica:** Donde sin que estrictamente se tenga que trabajar con muchísima data (podría ser tanto small Data como Big Data) se lleva a cabo análisis mayormente de tipo estadístico como análisis predictivos, construyendo modelos, un rol de esta vertiente sería la del Data scientist, esa persona encargada de hacer data mining, machine learning, etc.

1.3 TIPOS DE DATOS

Una vez hemos fijado con mayor precisión el concepto de Big Data, vamos a proceder a analizar los tipos de datos existentes, además de aclarar la diferencia entre lo que es Big Data y lo que son datos desde el punto de vista tradicional. Cuando las empresas deciden llevar a cabo un proyecto de Big Data deben dar solución a una serie de cuestiones tales como: el origen de los datos, el volumen de información necesario para tomar una decisión, la información que aporta cada dato a mi negocio... Por tanto, es importante que la empresa reconozca las fuentes de datos existentes y el tratamiento que necesita cada dato.

En Big Data los datos son diferentes a los datos tradicionales es decir los datos estructurados almacenados en bases de datos relacionales. Los datos se consideran en dos tipos, los estructurados y los no estructurados como podemos ver en la siguiente imagen:

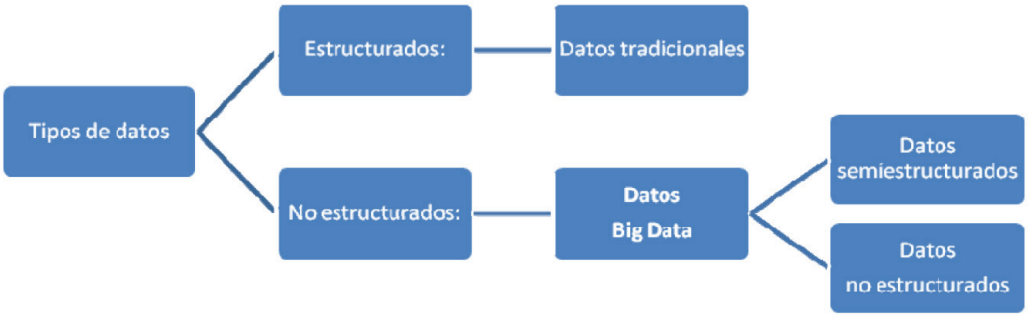

Figura 1.2. Tipos de datos en Big Data

- **Datos estructurados:** son aquellos datos con formato y campos fijos, en el que el formato es anticipadamente definido, para ser almacenados en bases de datos relacionales; este tipo de datos guardan un orden específico lo que facilita trabajar con ellos.
- **Datos semi estructurados**: son aquellos datos que no tienen formatos fijos, pero que contienen etiquetas, marcadores o separadores que permiten entenderlos; se procesan a base de reglas para extraer la información en piezas. Por ejemplo, los lenguajes XML y HTML son ejemplos de texto con etiquetas. no siguen un patrón claramente comprensible (como sí hacen los datos estructurados), a pesar de que, presentan un flujo claro y un formato definible. No existen formatos fijos como en los estructurados, pero sí marcadores para separar los datos. En esta categoría destacamos registros de logs procedentes de conexiones a internet.

- **Datos no estructurados:** son aquellos datos que no tienen formatos predefinidos, es decir no tienen estructura uniforme. Generalmente son datos binarios que no tienen estructura interna identificable. Es un conglomerado masivo y desorganizado de varios objetos que no tienen valor hasta que se identifican y almacenan de manera organizada. Por ejemplo los correos electrónicos, mensajes instantáneos SMS, WhatsApp, Viber, fotos, audios, videos entre otros. Su almacenamiento se da sin estructura uniforme y no existe capacidad para controlar estos datos. Los ejemplos más claros son los videos, audios, fotos o datos de texto (SMS, WhatsApp, Correos electrónicos...) Estos datos suponen el 80% de los datos que poseen las empresas, siendo con diferencia aquellos que presentan una mayor dificultad en su análisis, por tanto, han dado lugar al nacimiento de herramientas como MapReduce, Hadoop o bases NoSQL que analizaremos más adelante.

1.4 CARACTERÍSTICAS DE BIG DATA

Los últimos diez años han visto un aumento extraordinario del interés de empresas y organizaciones por el uso de herramientas que les permitan manejar la ingente cantidad de datos que recogen diariamente a través de sus sistemas de información, de sus canales de ventas y compras, de la información recogida a través de su presencia en la Web (anuncios, páginas de acceso a información, a servicios, etc.) o incluso cada vez más de comentarios y mensajes que se puedan generar en las redes sociales.

Este fenómeno ha incrementado enormemente la demanda de aplicación de procedimientos de análisis de datos para detectar la presencia de patrones o de tendencias que no resultan obvias, aportan información muy valiosa para mejorar significativamente su actividad: sus operaciones, sus ventas o sus resultados. Por otra parte, y asociado a este interés, se ha iniciado un proceso de revisión y mejora de las técnicas cuantitativas existentes para el tratamiento de datos y la extracción de la información relevante.

Uno de los aspectos más significativos asociado a este nuevo interés, y uno que resulta especialmente relevante por los cambios que implica tanto en la formación básica necesaria como en las aplicaciones para los profesionales interesados en el tratamiento de datos, es el aumento extraordinario en el volumen de los datos disponibles.

Cada vez es más habitual que las organizaciones y empresas dispongan de cantidades de datos medibles en peta- o exabytes (miles de billones o trillones de bytes).Se ha popularizado el uso del término “Big Data” para referirse a estas cantidades de información y a las técnicas adecuadas para su tratamiento. Un problema asociado a estos volúmenes de datos es que las técnicas tradicionales no resultan aplicables por ineficientes; es necesario utilizar nuevos métodos, adaptados especialmente a estas situaciones, creando una demanda y ofreciendo una oportunidad de formación de profesionales muy relevante en el futuro inmediato.

Tecnologías como Internet generan datos a un ritmo exponencial gracias al abaratamiento y gran desarrollo del almacenamiento y los recursos de red. El volumen actual de datos ha superado las capacidades de procesamiento de los sistemas clásicos de minería de datos. Hemos entrado en la era del Big Data o datos masivos, que es definida con la presencia de gran volumen, velocidad y variedad en los datos, tres características que fueron introducidas por D. Laney en el año 2001, con el requerimiento de nuevos sistemas de procesamiento de alto rendimiento, nuevos algoritmos escalables, etc.

IBM y Gartner plantean tres dimensiones para el entendimiento de la naturaleza de los Big Data, conocido como el modelo de las 3V; inclusive IBM considera una cuarta V correspondiente a la veracidad, y otras fuentes añaden una más, el valor. Sin embargo, las distintas fuentes tienen en común el modelo de las 3V que corresponde a volumen, velocidad y variedad. Otros dos

aspectos importantes que caracterizan los datos masivos son la veracidad de los datos y el valor intrínseco del conocimiento extraído. La figura muestra estas cinco características.

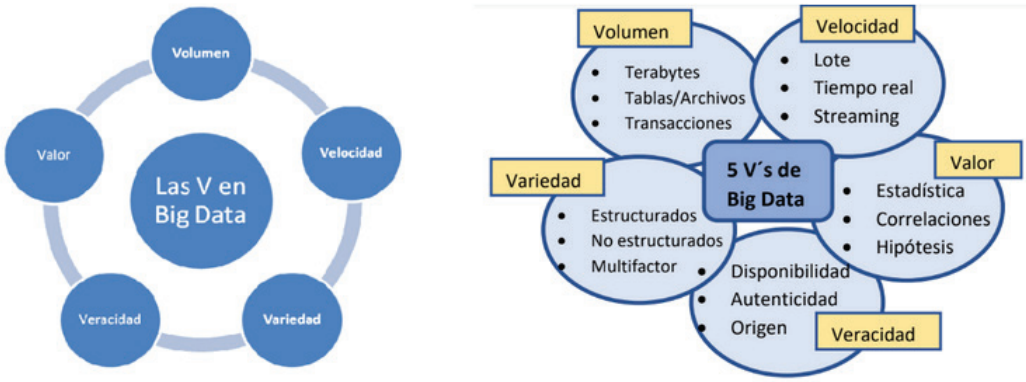

Figura 1.3. Características en Big Data

Dichos volúmenes de datos poseen cuatro características principales que vienen definidas como las cinco Vs:

- **Volumen de información.** Cantidad de datos que son generados a lo largo del tiempo. Es una de las principales características de Big Data, ya que hace referencia a las cantidades masivas de datos que se almacenan para ser procesados. Se espera que entre el presente año y el 2020, la humanidad entre en la era de zettabyte; en el año 2000 se almacenaron en el mundo 800.000 petabytes y se proyecta que para el año 2020 se alcancen 35 zettabytes (ZB); en 2012 Twitter generó más de 9 terabytes (TB) de datos al día. La gran cantidad de datos que las organizaciones generan es mayor día tras día, es por eso que hablar de un gran tamaño de datos en estos momentos es dar lugar a que solo en unos semestres ese valor no sea tan grande para ese momento. El volumen en Big Data hace referencia al presente de la cantidad de información que genera y almacena una organización, es decir a su volumen masivo de datos. Para IBM el volumen de datos disponible que tiene una organización está en ascenso en contraste con la disminución en el análisis que se hace de ellos.
- **Velocidad de los datos.** Rapidez con la que los datos son creados, almacenados y procesados en tiempo real. En muchas ocasiones es necesario hacer un estudio en tiempo real. En Big Data este tema merece consideración ya que el aumento de los flujos de datos en las organizaciones aumenta la velocidad en que se deben almacenar datos y sugiere últimas versiones de los gestores de grandes bases de datos. Este aumento en la velocidad de los datos requiere que el procesamiento de ellos se haga en tiempo real para mejorar la toma de decisiones.
- **Variedad de los datos.** Formas, tipos y fuentes en las que los datos son registrados. Los datos pueden ser estructurados y fáciles de gestionar como son las bases de datos, o no estructurados, como son los documentos de texto, correos electrónicos, datos de sensores, etc. Los datos no estructurados requieren un tratamiento diferente de los datos no estructurados, ya que es necesario un procesamiento de los datos recogidos de múltiples fuentes de información con la herramienta adecuada. representa todos los tipos de datos, entre datos estructurados y no estructurados. Las fuentes de datos son de cualquier tipo y ello aumenta la complejidad; por ejemplo los videos no pueden ser tratados en bases de datos relacionales como si se puede hacer con el registro de ingreso de los empleados

a una empresa. Considerar toda esta variedad de tipos de archivos supone un reto en la clasificación de los datos para que el análisis de ellos entregue resultados de valor a quien los investiga.

- **Veracidad de los datos**. Grado de falibilidad de los datos recibidos. Es necesario tener la certeza de que los datos obtenidos son de calidad, aplicando soluciones y métodos que puedan eliminar datos imprevisibles. En Big Data este término se relaciona a la fiabilidad de las fuentes de datos, debido al aumento de fuentes de ellos, y además a la variedad en los tipos de datos.
- **Valor**: Es la característica más importante de los datos. Como hemos visto anteriormente, el potencial de los datos es espectacular, pero de nada sirve tener acceso a una gran cantidad de datos si no somos capaces de convertirlos en algo con valor. Es decir, la información no sirve de nada a las organizaciones si esta no les otorga una fuente de valor, por tanto, para que las empresas realicen la inversión en almacenes de datos y sistemas de procesamiento y análisis debe existir un retorno claro de esta inversión. Como consecuencia de esto nace Hadoop cuya función es el análisis de grandes volúmenes de datos.

Es importante resaltar que, al pasar de la administración de una simple base de datos a adoptar el uso de Big Data, se necesita implementar una determinada arquitectura. Ésta viene marcada por el ciclo de vida del procesamiento de datos: capturar, organizar, integrar, analizar, actuar. En la siguiente imagen vemos los principales elementos cuando trabajamos con Big Data.

Figura 1.4. Elementos en Big Data

- **Collection (recogida):** una de las mayores dificultades a la hora de disponer los datos es cómo conseguirlos.
- **Storage (almacenamiento):** una vez han sido obtenidos, hay que determinar cómo almacenarlos de la manera más óptima para su gestión y posterior consulta.
- **Research (investigación):** la información que se pretende extraer de los datos debe ser parte de un proceso de investigación y de mejora continua para el descubrimiento de nuevas capacidades.
- **Analysis (análisis):** para que de los datos se pueda extraer una información valiosa, deben ser analizados.
- **Volume (volumen):** hablamos de Big Data y no de otras variaciones cuando se incluye un componente de volumen y complejidad.
- **Visualization (visualización):** para su mejor comprensión y sobre todo, de cara a poder orientar y convencer a los actores decisivos de una empresa, es imprescindible una visualización amigable del resultado del análisis.
- **Cloud technology (tecnología en la nube):** los datos deben estar disponibles para su consulta por distintos agentes en cualquier momento y desde distintas ubicaciones, además del hecho de que tener externalizados servicios en la nube tiene ventajas adicionales para una empresa, como se verá más adelante.
- **Network (red):** se trata de la infraestructura física que sustenta el punto anterior.

1.5 DESAFÍOS DE BIG DATA

Como toda tecnología en desarrollo, Big Data presenta desafíos relacionados a distintos factores, desde el hecho de hacer cambiar las infraestructuras y formas de pensar de los desarrolladores que hoy están acostumbrados a tecnologías como information retrieval y data mining, utilizando estilos tradicionales de desarrollo, hasta saber qué tipo de datos son los adecuados para buscar información para estas implementaciones. Entre los desafíos más comunes podemos citar los siguientes:

- **Skills:** Este problema trata básicamente la capacidad de las personas a cargo del manejo de la información recolectada. Al ser una tecnología en desarrollo, la cantidad de personas que tengan el "know how" o conocimiento para poder procesar de manera correcta el volumen de información es relativamente poco, lo que dificulta el desarrollo de proyectos.
- **Estructura de datos:** Otro gran desafío es la forma en la que se guardan los datos. La forma misma en que tenemos concebida la idea de cómo guardar los datos en la actualidad presenta un desafío enorme para Big Data. El desafío de hoy es que la mayoría de los almacenes de datos empresariales ven un cliente o una entidad que la empresa trabaja con una fila de datos en lugar de una columna. Esa fila se rellena y se actualiza quizás a diario con la instantánea o al agregado de la situación actual del cliente. Al realizar esta actualización, estamos perdiendo la información recolectada, lo que conlleva a menor capacidad de predicción o información a procesar.
- **La tecnología:** Lo interesante es que Hadoop es ideal para el procesamiento por lotes a gran escala, que es como las operaciones de agregación o cómputo. El problema es que Hadoop no es una tecnología en tiempo real o muy dinámica en absoluto. La ejecución de consultas en un clúster Hadoop suele tener una gran latencia ya que hay que distribuir cada consulta individual, luego, hacer su etapa de reducción, que está trayendo todos los datos de nuevo juntos. Así que es una tecnología de alto rendimiento, pero de alta latencia.

- **Privacidad:** Junto con la obtención de volúmenes de datos incalculables, viene una cantidad de datos que podríamos considerar intrusiva, podría darse ejemplos como Facebook, Twitter, Google que manejan grandes volúmenes de datos de clientes, con esta capacidad de Big Data de intentar analizar absolutamente todo, podría darse una examinación inapropiada de los datos de usuarios, conllevando rupturas en la privacidad de los datos de los usuarios. (Si bien esta problemática no es nueva, podría agravarse con la capacidad avanzada de procesamiento que se obtiene con Big Data).
- **Volumen, Variedad, Velocidad:** La capacidad de encontrar un equilibrio entre todas ellas depende de la capacidad de plantear un desarrollo sustentable y un plan acorde a las posibilidades tecnológicas de la empresa que desarrolla con esta tecnología.

A nivel técnico, la adopción de tecnologías big data supone una serie de desafíos entre los que podemos destacar:

- El análisis de datos estructurados es necesario para comprender los métodos de análisis de Big Data, incluso existen métodos que se comparten con el análisis convencional, pero con muchos más datos.
- La administración de bases de datos es un fundamento para el análisis de datos y para manejar datos operacionales. En Big Data, las bases de datos son una fuente importante que alimenta el núcleo de procesamiento.
- La programación orientada a objetos es el pilar para desarrollar cualquier tipo de aplicación, incluso para manejar bases de datos. El Big Data se utiliza para manejar y procesar distintos tipos de datos.
- La administración de servidores es necesaria para aprovechar al máximo las tecnologías de la información. En Big Data son primordiales pues son el soporte de toda la infraestructura de aprovechamiento de los datos masivos.

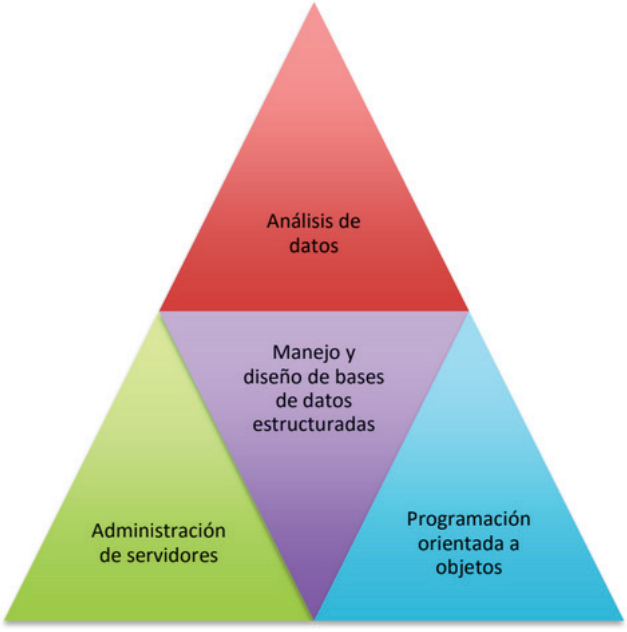

Figura 1.5. Desafíos en Big Data

1.6 TECNOLOGÍAS PARA BIG DATA

Las tecnologías y algoritmos sofisticados y novedosos son necesarios para procesar eficientemente lo que se conoce como Big Data. Estos nuevos esquemas de procesamiento han de ser diseñados para procesar conjuntos de datos grandes, datos masivos, dentro de tiempo de cómputo razonable y en un rango de precisión adecuado.

Desde el punto de vista del aprendizaje automático, esta problemática ha causado que muchos algoritmos estándar se conviertan en obsoletos en el paradigma Big Data. Como resultado surge la necesidad de diseñar nuevos métodos escalables capaces de manejar grandes volúmenes de datos, manteniendo a su vez su comportamiento en términos de efectividad.

Google diseñó MapReduce en 2003 la que es considerada como la plataforma pionera para el procesamiento de datos masivos, así como un paradigma para el procesamiento de datos mediante el particionamiento de ficheros de datos. MapReduce es capaz de procesar grandes conjuntos de datos, a la vez que proporciona al usuario un manejo fácil y transparente de los recursos del clúster subyacente.

En el paradigma **MapReduce**, existen dos fases: Map y Reduce. En la fase Map, el sistema procesa parejas clave-valor, leídas directamente del sistema de ficheros distribuido, y transforma estos pares en otros intermedios usando una función definida por el usuario. Cada nodo se encarga de leer y transformar los pares de una o más particiones. En la fase Reduce, los pares con claves coincidentes son enviadas al mismo nodo y finalmente fusionados usando otra función definida por el usuario. La siguiente figura muestra un esquema general del proceso completo MapReduce:

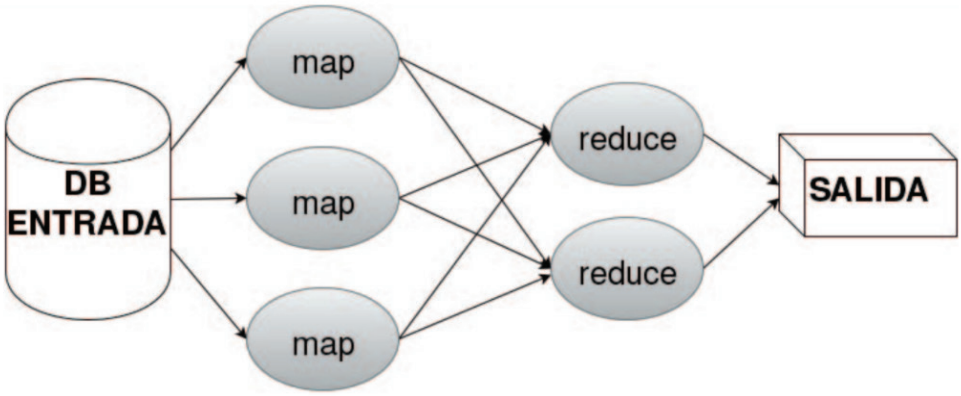

Figura 1.6. Modelo Mapreduce en Big Data

Este modelo consiste en dos funciones primitivas “Map” y “Reduce”. La entrada de “Map” es un conjunto de pares clave-valor (k1, v1) a los cuales se les aplica una función “Map” que devuelve como resultado un conjunto intermedio de pares clave-valor (k2 ,v2). Este conjunto intermedio se agrupa según claves iguales, las cuales sirven de entrada para la función “Reduce”, la cual trabaja sobre toda la lista de valores asociados a la misma clave y produce cero o más resultados agregados en forma de lista (list v3). Destacar que los conjuntos de pares clave-valor pueden pertenecer a dominios diferentes.

```
Map
Map(k1,v1) -> list(k2,v2)

Reduce
Reduce(k2, list (v2)) -> list(v3)
```

La función Map tiene como entrada una serie de pares <clave, valor> y produce una lista de pares intermedios como salida. La función Map, que internamente procesa los datos en cada proceso, es definida por el usuario siguiendo el esquema clave-valor. El esquema general para dicha función es el siguiente:

```
Map(<clave1, valor1>) -> lista(<clave2, valor2>)
```

En la segunda fase, el nodo maestro agrupa pares por clave y distribuye los resultados combinados a los procesos Reduce en cada nodo. La función de reducción es aplicada a la lista de valores asociada a cada clave y genera un valor de salida. Dicho proceso es esquematizado a continuación:

```
Reduce(< clave2, lista(valor2) >) →< clave3, valor3>
```

1.7 PERFILES BIG DATA

Un especialista en Big Data es un profesional que cuenta con amplios conocimientos en una serie de tareas involucradas en el ciclo de vida de la gestión de los datos tales como: identificar diversos orígenes de información, almacenar y extraer grandes volúmenes de datos, diseñar la arquitectura del ecosistema empresarial donde se procesa y consumirá los datos para su exploración, modelado, análisis, visualización y monitorización en tiempo real. Dependiendo de sus funciones, un especialista en Big Data debe poseer habilidades empresariales, técnicas y analíticas para obtener el mayor provecho de la información.

La constante y creciente generación de datos en todas las actividades humanas, y la consecuente necesidad de procesar y analizar un volumen cada vez mayor de información, implica una enorme oportunidad laboral. Un experto en Big Data forma parte de uno de los sectores profesionales con mayor oferta de empleos.

La clave para poder obtener, procesar, analizar y darles un aprovechamiento efectivo a los datos, pasa por la implementación de tecnologías adecuadas y contar con expertos en big data que sean capaces de gestionarlas e interpretar la información con foco en el negocio.

Dado que el uso de plataformas de Big Data aumenta cada vez más para dar paso a la transformación digital, es común que las empresas desarrollen sus propios sistemas con componentes legacy, en la nube o en ambos, por lo que los expertos de Big Data deben tener dominio en diferentes lenguajes de programación, aplicaciones tecnológicas, pero además de herramientas en entornos cloud.

Big Data con el panorama actual catapulta a los científicos de datos como otra muy buena opción de carrera profesional y sobre todo bien remunerada. Ya que el Big Data es una herramienta clave para las empresas para ganar competitividad, tomar decisiones basadas en datos.

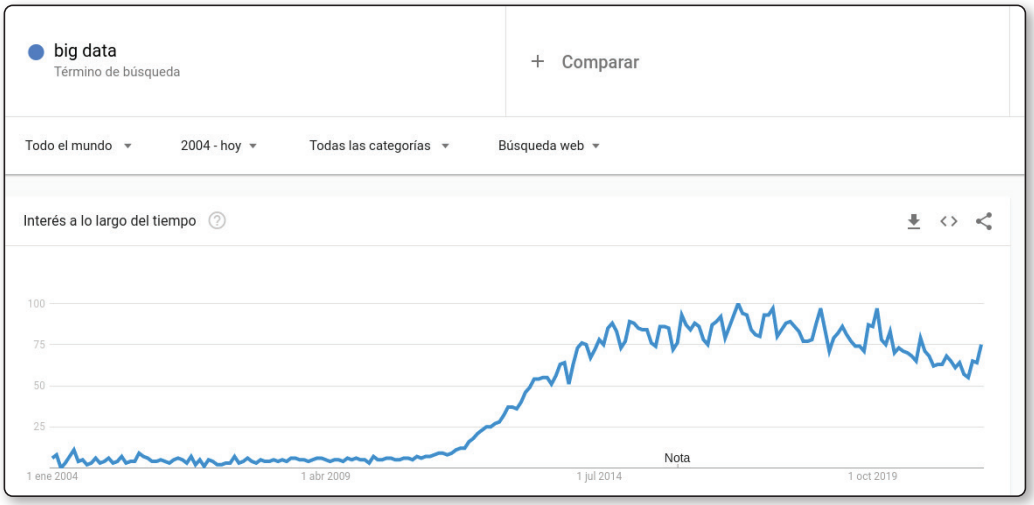

Figura 1.7. Evolución del término a lo largo del tiempo

Un aspecto muy importante es que los científicos de datos, no sólo se desarrollan como personas técnicas, es decir no están aislados en el área de sistemas y de allí no tienen interacción con el resto de la empresa a la que pertenecen, sino todo lo contrario, los científicos de datos van de la mano de la toma de decisiones de las empresas e interactúan con la mayoría de las áreas para obtener datos valiosos y saber cómo interpretarlos, es decir los científicos de datos están tomando decisiones o están al lado de los tomadores de decisiones.

Pero no solo eso se necesita para convertirse en un profesional de Big Data, además de tener algún máster o doctorado, se necesitan tener habilidades de comunicación ya que como se mencionó los científicos de datos tienen que estar en contacto con la mayoría de áreas de las empresas y por ende saber comunicarse con conocedores del dominio a tratar para sacar el mayor valor a los datos, se necesita un alto grado de curiosidad y tener una comprensión de lo que son negocios reales, deben de saber que una mala decisión tiene consecuencias reales en las empresas.

1.7.1 DIRECCIÓN DE DATOS(CHIEF DATA OFFICER-CDO)

Es el responsable de todos los equipos especializados en Big Data de la organización. Su función combina la rendición de cuentas y responsabilidad en cuanto a privacidad y protección de la información, calidad y gestión de los datos. Se trata del director digital de la empresa. Es una figura clave, ya que este profesional es el director digital de la empresa.

Se trata del líder de la gestión de datos y analítica asociada por el negocio, quien debe dirigir los equipos especializados en dato, definir políticas de seguridad para gestionar y almacenar datos, mantenerse actualizado en las regulaciones vigentes en cada país, decidir qué datos se utilizarán, incluyendo cómo y para qué, validar las tecnologías que se utilicen y ayudar a democratizar el acceso a los datos a todos los empleados y empleadas. Para aspirar a este puesto se requieren las siguientes **competencias**:

- Varios años de experiencia en el sector de la tecnología y trayectoria en el campo de la analítica aplicada al negocio.
- Formación en estadística y graduación en carreras como ingeniería, informática o telecomunicaciones. Se valoran los Másteres en Big Data, MBA o gestión de negocios.

- Habilidades de comunicación, planificación y gestión integral de proyectos, trabajo en equipo y marcación de objetivos.
- Capacidad analítica y orientación al cliente.

Este profesional es el encargado de coordinar los esfuerzos de todos los profesionales dedicados al Big Data en una organización. Debe establecer la metodología de trabajo, y asegurarse de que esta se encuentre enfocada en obtener los datos que la empresa necesita.

La formación profesional requerida para el cargo es la misma que requiere un experto en Big Data, pero generalmente para llegar a un puesto de CDO se requieren años de experiencia en el área. También se puede alcanzar este perfil combinando experiencia de Big Data con experiencia a nivel de gestión.

1.7.2 CIENTÍFICO DE DATOS(SCIENTIST)

El científico de datos analiza, interpreta y comunica las nuevas tendencias en el área y las traduce a la empresa para que puedan hacer uso de ellas y así adaptar sus productos y servicios y crear nuevas oportunidades de negocio. Es el encargado de traducir la información para que los analistas puedan tomar decisiones.

Para el perfil de científico de datos se precisan conocimientos estadísticos que un programador no suele tener y conocimientos informáticos que un estadístico no suele manejar. Dentro de este perfil diferenciamos entre los profesionales orientados al campo de las matemáticas y las estadísticas y los que proceden del ámbito de la inteligencia artificial y el machine learning. Este perfil debe unir conocimientos de matemáticas, estadística y programación, y conocer también muy al detalle el sector de actividad de la compañía para la que trabaja, además de ser buen comunicador para trasladar los datos que interpreta.

La principal función del científico de datos es la de traducir los grandes volúmenes de datos y convertirlos en información útil para la empresa. Tiene conocimientos matemáticos, estadísticos y de programación. También cuenta con una visión de negocio y habilidades comunicativas, para dar a conocer el resultado de su trabajo al resto de la organización.

Permiten extraer conocimiento e información valiosa de los datos. Tienen visión general del proceso de extremo a extremo y pueden resolver problemas de ciencias datos, la construcción de modelos analíticos y algoritmos. Combinan diversas habilidades relacionadas con las matemáticas, la estadística, la programación y visualización, pero también deben tener habilidades comunicativas, para explicar los resultados obtenidos en la organización. Estas disciplinas están en línea con las habilidades que se demandan hoy en día de un **data scientist**:

- **Programación**: Para la limpieza, tratamiento, filtrado, etc. de los datos es necesario conocimientos de Programación.
- **Informática**: Nos dará la infraestructura y herramientas necesarias para almacenar los datos, procesarlos, etc., especialmente cuando nos movemos en el mundo Big Data.
- **Estadística**: Para la obtención y visualización de insights, responder las cuestiones planteadas, representar la información que obtengamos, ... saber qué modelos, algoritmos, etc. podemos utilizar, cómo validar los resultados, …
- **Matemáticas**: Para entender los fundamentos de los modelos y técnicas estadísticas que empleemos.

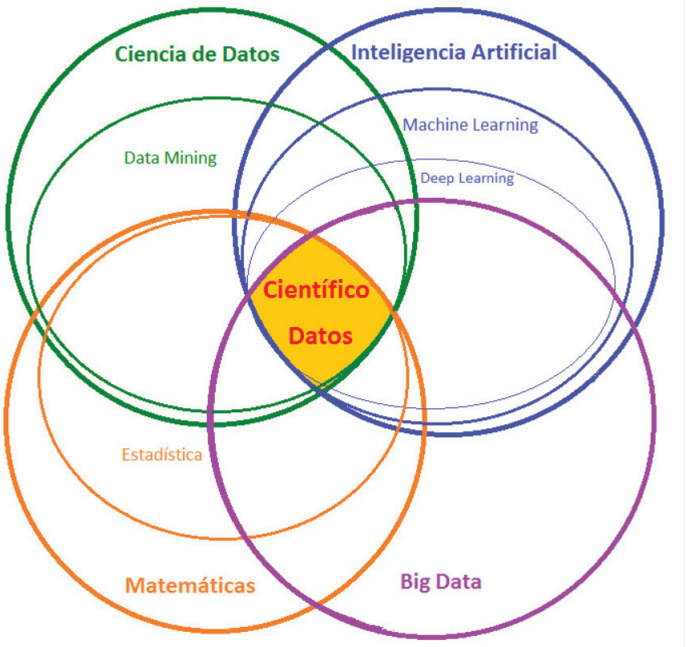

Figura 1.8. Áreas de conocimiento de un científico de datos

Se trata de un perfil muy buscado por empresas vinculadas a motores de búsqueda, servicios financieros y e-commerce, ya que su aporte reside en la extracción de información valiosa de los datos generados en el marco de la operación, con una visión general del proceso. Para aspirar a este puesto se requieren las siguientes **competencias**:

- Conocimientos de ingeniería de software en sistemas distribuidos, algorítmica y estructuras de datos.
- Ser experto en matemáticas, estadística, informática, etc.
- Saber de Machine Learning, lenguajes de programación como R o Python, y el uso de notebooks y ecosistemas Big Data.
- Poseer gran capacidad para la resolución de problemas.
- Capacidad para analizar, resolver y explicar en forma entendible evitando conceptos científicos, y predecir comportamientos futuros.
- Conocimientos en programación y aptitud para expresarse en lenguajes informáticos.
- Facilidad en álgebra lineal, cálculo y probabilidad.
- Comprensión y manejo de las técnicas de machine learning.
- Pensamiento lógico, análisis, predicciones y capacidad de detección de patrones.
- Capacidad para usar librerías como Tensor Flow para técnicas de Deep Learning basadas en redes neuronales.

Básicamente, su trabajo consiste en analizar y gestionar la información generada por los usuarios, y transformarla en datos comprensibles para las empresas. Mediante la creación de gráficos y estadísticas, este profesional será capaz de arrojar luz sobre miles de millones de datos en bruto.

Para alcanzar este perfil, lo recomendable es optar por carreras de Informática o Tecnología y luego realizar una especialización. Aunque existen instituciones privadas que ofrecen cursos breves con los que se puede adquirir este perfil.

Este perfil presenta un nivel de conocimientos superior al del analista de datos, pudiendo realizar sus mismas tareas, pero con un nivel de profundidad superior a nivel de conocimientos matemáticos y de programación, que les permiten conocer los detalles de implementación de los métodos y algoritmos de minería de datos y machine learning.

Los científicos de datos se dedican a resolver problemas con casuísticas complejas, muchas veces de problemas ad hoc que requieren un análisis y dedicación profunda. Deben de ser capaces de hacer investigación y conocer el estado del arte en los temas de minería de datos y machine learning, ya que la optimización de los algoritmos mediante la parametrización (fine-tuning) es una de sus responsabilidades.

1.7.3 ANALISTA DE DATOS(DATA ANALYST)

El perfil de analista de datos se encuentra en la intersección de otras disciplinas como Informática y Programación, Estadística y Matemáticas. Como su cargo indica, se encarga de participar en el análisis de los datos y recoge las necesidades de los clientes para presentarlas al Data Scientist. También se encarga de extraer, procesar y agrupar datos, analizar esas agrupaciones de datos y generar informes.

Es uno de los perfiles profesionales más demandados actualmente por las empresas ya que se encarga de procesar la información y obtener conclusiones que ayuden a mejorar resultados. Estos profesionales son los que saben extraer patrones de conducta de los usuarios y saben por qué actúan de una manera.

Tiene la responsabilidad de descubrir cómo extraer datos, procesarlos y sintetizarlos para obtener conclusiones y resolver aquellos problemas surgen en una organización, a través de modelos computarizados avanzados, y modelos analíticos y de visualización de datos sintonizados con los requerimientos de una compañía. Para aspirar a este puesto se requieren las siguientes competencias:

- Estudios de grado en Estadística, Matemáticas o Ingenierías (técnica y/o superior).
- Dominio de lenguajes de programación como Python, y programas estadísticos.
- Capacidad para extraer, limpiar, analizar, modelar e interpretar datos.
- Habilidades de comunicación, planificación y trabajo en equipo.
- Además de los conceptos de Machine Learning, deben destacarse por el conocimiento del entorno Big Data en el que trabajan, como Spark o Hadoop.
- Son valorados los conocimientos de Bases de Datos SQL y Business Intelligence.

1.7.4 INGENIERIO DE DATOS(DATA ENGINEER)

La principal tarea de un ingeniero de datos es la de distribuir datos de manera accesible a los Data Scientist. Su perfil es más especializado en gestión de bases de datos y en sistemas de procesamiento y de programación. Podríamos definir un Data Engineer como aquel profesional

enfocado en el diseño, desarrollo y mantenimiento de los sistemas de procesamiento de datos dentro de un proyecto de Big Data. Entre las principales que desempeña podemos destacar:

- Proporcionar los datos de una manera accesible y apropiada a los usuarios y Data scientists.
- Desarrollar y explotar técnicas, procesos, herramientas y métodos que deben servir para el desarrollo de aplicaciones Big Data.
- Tiene un gran conocimiento en gestión de bases de datos, arquitecturas de clusters, lenguajes de programación y sistemas de procesamiento de datos.
- Construir y mantener las estructuras y las arquitecturas tecnológicas necesarias para el procesamiento, ingestión e implementación a gran escala de aplicaciones que usan datos de forma intensiva.
- Se especializa en infraestructura Big Data, creando e implementando técnicas, procesos, herramientas y métodos para el desarrollo de aplicaciones Big Data.
- Juega un papel clave a la hora de convertir una prueba de concepto de Big Data en un proyecto real y palpable.

Para aspirar a este puesto se requieren las siguientes **competencias**:

- Conocimiento en gestión de bases de datos, arquitecturas de clusters, lenguajes de programación y sistemas de procesamiento de datos.
- Trabajar con Linux y Git, y también con Hadoop y Spark a nivel de entornos, Mapreduce a nivel de modelos computacionales, y HDFS, MongoDB y Cassandra a nivel de tecnologías NoSQL.
- Se suelen requerir los siguientes lenguajes: Python para el procesado de datos con librerías como **PySpark** y Scala como lenguaje nativo de Spark y Java, en muchos casos.

1.7.5 ARQUITECTO DE DATOS(DATA ARCHITECT)

El Arquitecto de Big Data es quien define la infraestructura de la plataforma de Big Data. Tiene una visión global tanto de las necesidades de las empresas u organizaciones como de las soluciones de tratamiento de datos recomendadas para cada caso.

Cuando el ingeniero de datos se dedica a diseñar, implementar y mantener infraestructura relacionada con sistemas de Big Data, se llama arquitecto de Big Data. En este caso, su trabajo se centra en el diseño, creación y mantenimiento de clusters de procesamiento distribuido, como por ejemplo Apache Hadoop, Apache Spark o Apache Flink, y sistemas de almacenamiento distribuido de datos, como por ejemplo el sistema de ficheros distribuido HDFS o las bases de datos NoSQL.

Este perfil tiene como objetivo velar por el buen funcionamiento y la seguridad de las plataformas y el hardware que contienen los datos, debiendo prever los nuevos escenarios de volumen de datos que se puedan presentar en un futuro. Para aspirar a este puesto se requieren las siguientes **competencias**:

- Formación en Informática y/o matemáticas.
- Experiencia para manejar tecnologías de datos no estructurados como Hadoop, Spark o Cassandra.
- Conocimientos de lenguajes de programación distribuida y funcional como Java/scala, SQL y Python.
- Conocimientos en bases de datos como Oracle y PostgreSQL.

Estos perfiles tienen una formación en ingeniería informática, matemática, física o telecomunicaciones. Algunas de las herramientas que deben manejar son Hadoop, MapReduce, Hive, Pig, Spark, Flink, experiencia en SQL (MySQL, PostgreSQL) y NoSQL (Hive, Couchbase, Redis, ElasticSearch, Solr), conocimientos avanzados en Java, Scala o Python o conocimientos avanzados de estructuras de datos, data mining aplicado y machine learning.

En los casos de Data Scientist provenientes del mundo de la estadística suelen trabajar con R como herramienta principal y tratan de realizar las tareas de manipulación y agregación de datos mediante R, aunque a veces en el mundo Big Data no sea la solución ideal.

Sin embargo, en los casos de Data Scientists que vienen del mundo del desarrollo de software, estos profesionales se sienten más cómodos con lenguajes más formales, y aquí es donde Python juega un papel fundamental, las distintas contribuciones, módulos y librerías de análisis, de machine learning y librerías específicas (series temporales, procesado de lenguaje natural, ...) junto con su fácil integración con las plataformas Big Data hacen de Python la tecnología ideal para análisis de datos.

1.7.6 GESTOR DE DATOS(DATA MANAGER)

El objetivo principal de los gestores de datos o Data Manager es supervisar los diferentes sistemas de datos de una empresa. Son los encargados de organizar, almacenar y analizar los datos de la forma más eficiente posible.

Los Data Manager tienen conocimientos relacionados con la informática y de uno a cuatro años de experiencia. Destacan en el mundo de los números, los registros y los datos en bruto. Pero también tiene que estar familiarizado con todo el sistema de datos y tener una mente lógica y analítica con buena capacidad para resolver problemas.

1.7.7 CIUDADANO CIENTÍFICO DE DATOS(CITIZEN DATA SCIENTIST)

Es el profesional que no tiene una formación específica en Data Scientist, pero que con su experiencia puede aportar valor. Por ejemplo, realizando tareas analíticas y de gestión de datos a través de herramientas más sencillas. Se define como una persona que crea o genera modelos que utilizan análisis de diagnóstico avanzado o capacidades predictivas, pero cuya función principal de trabajo está fuera del campo de la estadística y el análisis.

En resumen, son personas no técnicas que pueden usar herramientas de ciencia de datos para resolver problemas relacionados con big data. Su experiencia y su conocimiento de las prioridades de la organización les permiten integrar de forma eficaz la ciencia de datos y el desarrollo de machine learning en los procesos.

1.7.8 ADMINISTRADOR DE DATOS(DATA STEWARD)

Este especialista es el responsable de velar por la calidad, la seguridad y la disponibilidad de los datos. Su función se resume en saber utilizar los datos dentro del proceso de negocio y presentarlos a toda la organización.

Este perfil trata la gestión y supervisión de los activos de datos de una organización para ayudar a proporcionar a los comerciales datos de alta calidad a los que poder acceder fácilmente. Data Steward se enfoca en la coordinación e implementación de tácticas. Es responsable de llevar a cabo el uso de datos y las políticas de seguridad según lo determinado a través de iniciativas empresariales, actuando como enlace entre el departamento de IT y el comercial.

1.7.9 TABLA COMPARATIVA

Como se puede observar en la siguiente tabla la diferencia entre algunos de los roles es cuestión de matices.

Ingeniero de datos	Perfil orientado al desarrollo de software y con experiencia en el tratamiento de datos desde la extracción y depuración hasta el procesamiento y el almacenamiento.
Arquitecto Big Data	Encargado de definir la arquitectura de los sistemas Big Data, eligiendo las alternativas más óptimas desde el punto de vista de seguridad, gobierno del dato y rendimiento. También es el responsable de mantener las plataformas actualizadas tecnológicamente y proponer nuevas alternativas que mejoren lo existente cuando van apareciendo.
Científico de datos	Perfil que cuenta con background de investigación en ámbitos de ingeniería, física y estadística. Experto en tratar problemas complejos, extrapola el conocimiento adquirido en el contexto académico para resolver problemas planteados en el entorno empresarial.
Analista de negocio	Profesional orientado al negocio con capacidad para comprender los resultados derivados del análisis avanzado de datos. Crea propuestas de valor para el negocio con el fin de generar beneficios para la empresa.
Ingeniero de visualización de datos	Perfil diferencial en visualización de datos y storytelling con capacidad para explotar el valor de los datos y hacerlos entendibles. Aplica herramientas de programación, de Data Discovery y de visualización.

El rol del Científico de Datos es el más importante en cuanto a la interpretación de los datos, diseño de algoritmos y análisis predictivos, es el que aplica métodos matemáticos y estadísticos a los datos para obtener valor de ellos, adicionalmente aplica conocimientos y metodologías de distintas áreas a los datos como machine learning, deep learning, inteligencia artificial.

En cuanto al Ingeniero de Datos es quien diseña e implementa la solución de Big Data para almacenar, consumir, analizar, visualizar los datos. También es el encargado de decidir qué tecnologías se adaptan mejor a la situación que se está tratando para obtener el mayor beneficio y valor de los datos. Al estar relacionado con el desarrollo, suele tener conocimiento sobre lenguajes de programación orientados a análisis de datos como R y Python.

1.8 BIG DATA ANALYTICS

En la actualidad los Big Data pueden ayudar a responder cuestiones clave acerca de cómo se comportan los clientes, cómo van a funcionar los nuevos lanzamientos, las futuras campañas o las posibles promociones. Esto está contribuyendo a realizar mejoras en los negocios mediante el marketing personalizado (one-to-one), las estrategias de competencia monopolística en precios, el análisis de atribución para estímulos comerciales, etc.

Por este motivo, se suele considerar que Big Data, más que tratar sobre datos, trata "sobre la transformación empresarial, sobre pasar del planteamiento retrospectivo de la monitorización y el procesamiento de datos por lotes a la obtención de conocimientos empresariales en tiempo

real". La Era del Big Data, por tanto, produce una creciente competencia en la comprensión de las necesidades del cliente en todo momento.

La posibilidad de aplicar técnicas de Big Data Analytics está haciendo que las empresas introduzcan paulatinamente una mayor "cultura de los datos" ("Data-driven culture") dentro de su operativa empresarial habitual, recurriendo tanto a nuevas tecnologías de almacenamiento y gestión de datos, como a herramientas de visualización y monitorización de métricas acerca del funcionamiento de la empresa (Key Performance Indicators - KPI) insertas en cuadros de mando ("dashboards").

Los indicadores clave de desempeño o KPI son valores que indican el rendimiento de un proceso de acuerdo con un objetivo predeterminado. Toda organización debe ser capaz de identificar sus propios KPI, por lo que deben tener:

- Definido completamente y acotado su proceso de negocio.
- Objetivos claros o el rendimiento del proceso de negocio.
- Una medida cuantitativa o cualitativa de los resultados con relación a los objetivos.
- Información sobre las variaciones entre los resultados y los objetivos planteados para ajustar procesos o recursos y alcanzar metas a corto plazo.

De este modo, la era del Big Data está haciendo que las empresas evolucionen por los estados de madurez siguientes: en primer lugar, la analítica descriptiva, en la que únicamente se dispone del dashboard en estado inicial; en segundo lugar la analítica de diagnóstico, enfocada a una comprensión avanzada y continua de la situación empresarial; en tercer lugar, la analítica predictiva, enfocada en la anticipación de riesgos y oportunidades; en cuarto lugar, la analítica prescriptiva, enfocada a la recomendación de acciones; y, por último, la analítica cognitiva, hoy en día emergente.

La analítica de Big Data es el proceso de examinar con gran velocidad, conjuntos de grandes volúmenes de datos entre una amplia variedad de tipos y descubrir patrones ocultos, nuevas correlaciones y más información útil, en un tiempo razonable en el que la oportunidad de la información proporcione ventajas competitivas al investigador.

Los grandes volúmenes de información pueden proceder de fuentes de datos no estructurados como los que generan smartphones, medios de comunicación, información suministrada por sensores, actividades sociales, entre otros; pero, además pueden proceder de datos estructurados almacenados en bases de datos relacionales.

El análisis de grandes datos (analítica de Big Data o Big Data analytics) corresponde a datos estructurados, no estructurados y semiestructurados. El análisis de grandes datos relacionales se puede realizar con herramientas de software tradicionales con técnicas sencillas o avanzadas como minería de datos, análisis predictivo y análisis estadísticos.

En cuanto a las fuentes de datos no estructuradas, pueden no encajar dentro de los esquemas de los almacenes de datos tradicionales o EDW (Enterprise Data Warehouse) o no estar en capacidad de atender la demanda de procesamiento de datos requerido.

Para atender la demanda de procesamiento de grandes datos han surgido tecnologías de bases de datos distintas a las relacionales llamadas bases de datos NoSQL, bases de datos en memoria y MapReduce. Este sistema se integra a través de un cluster, y bien puede ser por medio de software de código abierto o propietario. Para el tratamiento de los grandes volúmenes de datos se requieren las siguientes etapas:

- **Adquisición de datos:** los datos proceden de fuentes de datos diversas, es decir, de fuentes de datos tradicionales como almacenes de datos, bases de datos relacionales, entre otros; y, de fuentes de datos no estructurados o semi estructurados. Los datos procedentes de ambos tipos de fuentes de datos, pueden ser almacenados en bases de datos NoSQL o en bases de datos "en memoria".
- **Organización de los datos**: el origen distinto en las fuentes de datos requiere que luego de que se adquiera la información, deba prepararse, siendo tal vez necesario eliminar datos o parte de ellos para dejar lo más relevante de estos.
- **Análisis de información**: es una etapa muy importante dentro del tratamiento de los grandes volúmenes de datos. Consiste en analizar todos los datos por medio de herramientas estadísticas avanzadas como minería de información, minería social, herramientas desarrolladas para diseño de estadística avanzada como el lenguaje de programación R.
- **Decisión**: es en esta etapa en donde con los resultados obtenidos del análisis de información se obtiene conocimiento, preferiblemente en tiempo real; para que se incluya en los tableros de control, cuadros de mando y herramientas de visualización, y así predecir el comportamiento que va a tener el objeto de estudio.

El **preprocesamiento de datos** es una etapa fundamental en el proceso de extracción de conocimiento, cuyo objetivo principal es obtener un conjunto de datos final que sea de calidad y útil para la fase de extracción de conocimiento. El preprocesamiento de datos se vislumbra como una herramienta muy importante en el paso de Big Data a Smart Data, esencial para convertir los datos almacenados (material en bruto) en datos de calidad (valga el símil del paso de un diamante en bruto sin pulir y sin tallar a la piedra preciosa tras su procesado).

Para la mayoría de problemas actuales con datos masivos es necesario el uso de una solución distribuida escalable porque las soluciones secuenciales no son capaces de abordar tales magnitudes. Varias plataformas para el procesamiento a gran escala (como Spark o Hadoop) han intentado afrontar la problemática del Big Data en los últimos años. Estas plataformas requieren algoritmos escalables que den soporte a las tareas más relevantes de la analítica de datos masivos.

Los algoritmos de preprocesamiento también están afectados por el problema de la escalabilidad, por lo tanto deben ser rediseñados para su uso con tecnologías Big Data si queremos preprocesar conjuntos de datos masivos en los diferentes escenarios de aplicación, aprendizaje supervisado y no supervisado, procesamiento en tiempo real (flujo masivo de datos), etc.

Los que se llaman modelos no supervisados, en los que se incluyen las Reglas de Asociación, Patrones Secuenciales y Clustering. Los modelos supervisados, que necesitan un conjunto de entrenamiento del que aprender y que se suele etiquetar manualmente. Aquí se incluyen los modelos que pretenden adivinar un valor numérico para la variable objetivo que se quiere adivinar (predicción) o una etiqueta (clasificación) que puede tener únicamente dos valores posibles (binaria) o más de dos (multiclase).

La preparación de datos está formada por una serie de técnicas que tienen el objetivo de inicializar correctamente los datos que servirán de entrada para los algoritmos de minería de datos. Este tipo de técnicas pueden clasificarse como de uso obligado, ya que sin ellas los algoritmos de extracción de conocimiento no podrían ejecutarse u ofrecerían resultados erróneos. En esta área se incluye la transformación de datos y normalización, integración, limpieza de ruido e imputación de valores perdidos.

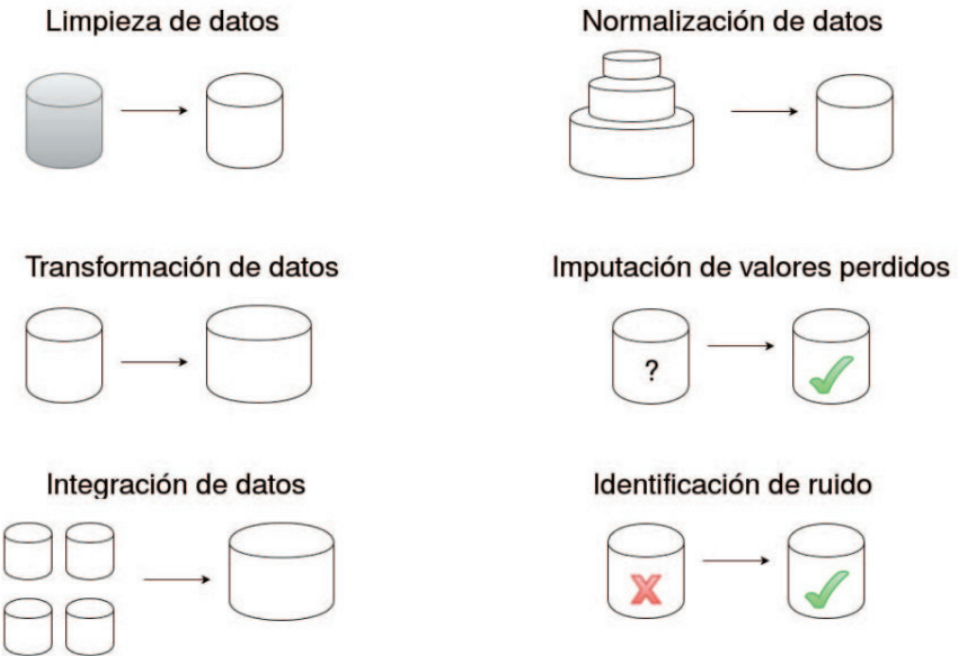

Figura 1.9. Etapas relacionadas con la preparación de los datos

Las técnicas de reducción de datos se orientan a obtener una representación reducida de los datos originales, manteniendo en la mayor medida posible la integridad y la información existente en los datos. Cuando el tiempo de ejecución de un algoritmo o el tamaño de los datos comienza a ser bastante elevado, para los algoritmos de extracción, estas técnicas deben ser aplicadas para obtener conjuntos de datos más pequeños y de calidad. En esta área las técnicas de reducción más relevantes son:

- **Selección de atributos (Feature Selection)**. El objetivo es reducir el número de atributos iniciales para reducir la complejidad a la hora de realizar el análisis.

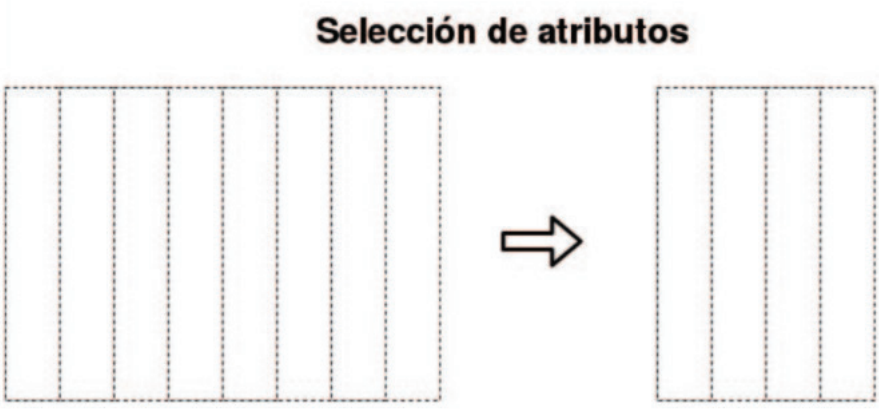

Figura 1.10. Selección de atributos

- **Selección de instancias (Instance Selection)**. El objetivo es reducir el número de filas que contiene nuestro dataset inicial.

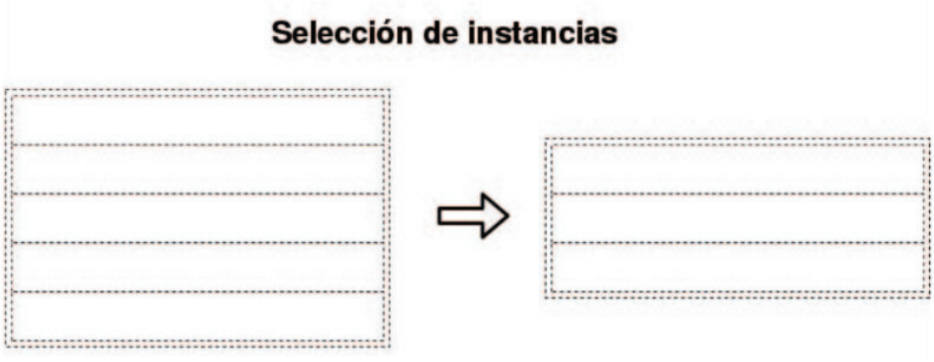

Figura 1.11. Selección de instancias

- **Discretización**. El objetivo es convertir variables numéricas en variables categóricas que nos permitan realizar una clasificación de los posibles valores que puede tomar esa variable.

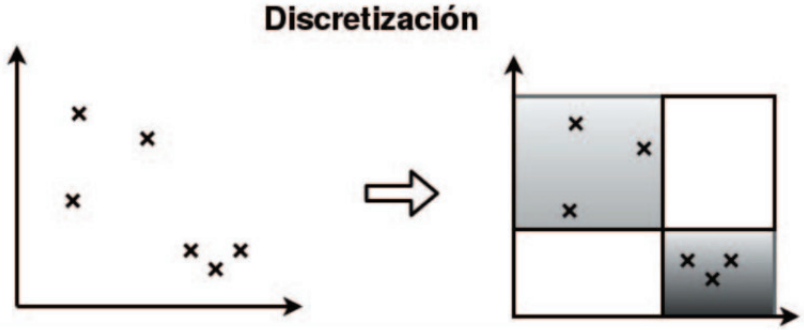

Figura 1.12. Discretización

2

ARQUITECTURAS BIG DATA

2.1 INTRODUCCIÓN

Ya sabemos en qué consiste Big Data, y que dentro de sus 5V, dos de las más importantes son el volumen y la velocidad. Para cumplir con estas necesidades, necesitamos una infraestructura que dote a nuestras aplicaciones de toda la potencia y robustez necesarias. La solución a estas necesidades aportará asimismo varias oportunidades:

- Incrementar la cantidad de datos que se gestionan.
- Incluir en los sistemas el tratamiento masivo e inteligente de datos no estructurados.
- Generar nuevos indicadores, reflejo de nuevas capacidades generadas.
- Tratamiento analítico de datos en tiempo real.
- Generación de información predictiva y prospectiva que puede ser integrada en otros sistemas.

Una arquitectura Big Data se define como un sistema de tratamiento de datos creado para tomar inputs de diferentes fuentes y con distintos formatos, analizarlos y convertirlos en conclusiones relevantes para el destinatario, de manera que le ayuden a predecir escenarios o determinar qué acción llevar a cabo en una situación dada.

Las arquitecturas Big Data se basan tanto en el almacenamiento como en el procesamiento distribuido de los datos, lo que las hace más seguras que los modelos centralizados en caso de fallos, ya que facilitan la localización y el aislamiento del nodo problemático con celeridad sin comprometer el funcionamiento del resto. Esto reduce, asimismo, la latencia en las conexiones, acortando los tiempos de respuesta en las solicitudes de información.

Otra de las grandes ventajas de las arquitecturas Big Data con respecto a las metodologías de análisis tradicionales es su escalabilidad ya que se conciben como sistemas adaptativos, preparados para asumir la entrada continua de nuevos conjuntos de datos y para ser extrapolados a ámbitos muy diversos.

Una arquitectura de big data se diseña para manejar la ingestión, el procesamiento y el análisis de los datos que son demasiado grandes o complejos para un sistema tradicional de base de datos. Todas las arquitecturas que diseñemos deberían cumplir las siguientes características:

- **Escalabilidad**: permite aumentar fácilmente las capacidades de procesamiento y almacenamiento de datos.
- **Tolerancia a fallos**: garantiza la disponibilidad del sistema, aunque se produzcan fallos en algunas de las máquinas, evitando la pérdida de datos.
- **Datos distribuidos**: los datos deben estar almacenados entre diferentes máquinas evitando así el problema de almacenar grandes volúmenes de datos en un único nodo central.

- **Procesamiento distribuido:** el tratamiento de los datos se realiza entre diferentes máquinas para mejorar los tiempos de ejecución y dotar al sistema de escalabilidad.
- **Localidad del dato:** los datos a trabajar y los procesos que los tratan deben estar cerca, para evitar las transmisiones por red que añaden latencias y aumentan los tiempos de ejecución.

2.2 ACTORES PRINCIPALES EN UNA ARQUITECTURA BIG DATA

Dentro de una arquitectura Big Data existen diversos actores que cumplen roles de gestión, operación, procesamiento, seguimiento y seguridad, entre otros, además de interactuar entre ellos o con otros componentes externos e internos de la arquitectura.

Para que los roles puedan cumplir un funcionamiento óptimo dentro de una arquitectura Big Data se debe prestar especial atención a la validación de los datos. Si para el proyecto se van a tener una combinación de fuentes de datos, es fundamental el poder validar que estas fuentes tienen sentido por sí mismas o cuando se combinan. Además, ciertas fuentes de datos, sobre todo los datos médicos, pueden contener información sensible por lo que se debe poner en práctica un nivel suficiente de seguridad y gobernabilidad.

Por supuesto, el diseño de una arquitectura Big Data en primer lugar tiene que comenzar con la definición del problema para luego establecerse ciertas características como el tipo de datos o combinación de varios de ellos, si es necesario el procesamiento en tiempo real o un procesamiento por lotes, por mencionar algunas consideraciones a tomar en cuenta.

Figura 2.1. Tecnologías Big Data

A continuación, se pasa a detallar las características principales de cada componente de una posible arquitectura big data y los roles de cada uno de ellos dentro de la arquitectura.

2.2.1 SISTEMA DE ORQUESTACIÓN

El rol de orquestación proporciona los requisitos del sistema y la monitorización del sistema de datos. Este implica una colección de funciones más específicas, que pueden ser ejecutadas por uno o más actores, que gestionan y orquestan la operación del sistema Big Data. También se incluyen las tareas de supervisión o auditoría para asegurar que el sistema cumple con dichos requisitos.

Estos actores pueden ser usuarios, componentes de software o alguna combinación de ambos. Las cargas de trabajo administradas por este componente pueden ser asignadas a nodos

físicos o virtuales a un bajo nivel o proporcionar una interfaz gráfica de usuario donde se pueda realizar la especificación de los flujos de trabajo y que además integre múltiples aplicaciones y componentes de alto nivel.

2.2.2 PROVEEDOR DE DATOS

Este rol hace que los datos estén disponibles para sí mismo como para otros roles. Para cumplir con su función, crea una abstracción de diversos tipos de fuentes de datos (como datos en bruto o datos previamente transformados por otro sistema) y los pone a disposición a través de diferentes interfaces funcionales.

El actor que desempeña este rol puede formar parte del sistema Big Data de manera interna o externa a la organización y pueden ser desde un sensor, un usuario que introduce datos de forma manual, a otro sistema Big Data. Las fuentes de datos pueden ser registros internos o públicos, audio, imágenes, vídeos, datos de sensores, web logs, registros de auditoría, cookies HTTP, entre otras fuentes. El proveedor de datos incluye actividades que son comunes a la mayoría de sistemas de gestión de datos entre las que podemos destacar:

- Recolección y persistencia de los datos.
- Proporcionar funciones de transformación para la depuración de datos confidenciales.
- Creación de metadatos que describen las fuentes de datos, políticas de acceso y otros atributos relevantes.
- Hacer que los datos sean accesibles a través de interfaces intuitivas donde la información sea requerida o solo enviada por el sistema sin necesidad de hacer una solicitud.
- Publicar la disponibilidad de la información y los medios para acceder a ella.

2.2.3 PROVEEDOR DE APLICACIONES BIG DATA

El proveedor de aplicaciones Big Data lleva a cabo un conjunto de operaciones en el ciclo de vida de los datos para cumplir con los requisitos establecidos por el sistema de orquestación.

El proveedor de aplicaciones de Big Data puede ser una sola instancia o una colección de estas, que se ejecutan en las diferentes etapas del ciclo de vida de los datos. Este rol consta de actividades que son representadas como subcomponentes, las cuales se detallan a continuación:

- **Recolección de datos.** La actividad de recolección de datos está integrada con el componente proveedor de datos. Esta puede ser un servicio como un servidor de archivos o un servidor web configurado por el sistema de orquestación para realizar recolecciones particulares de datos, también puede ser una aplicación diseñada para extraer o recibir datos desde el proveedor de datos. La persistencia de los datos puede no ser necesaria debido a que se pueden utilizar procesamientos en memoria u otros servicios proporcionados por el proveedor de infraestructura Big Data.
- **Preparación**. En la actividad de preparación es donde se podría indicar que se lleva a cabo el proceso de transformación de un ciclo ETL (Extract, Transform, Load), aunque la actividad de análisis también puede ser considerada como la ejecución de tareas avanzadas de la transformación. Entre las tareas que realiza esta actividad se pueden incluir la validación de datos, limpieza, normalización, entre otras.
- **Análisis**. La actividad implementa los métodos y técnicas para extraer conocimiento de los datos basados en los requisitos del sistema. Por lo general, esta actividad proporciona software para el análisis en streaming o por procesamiento por lotes. La plataforma de comunicación y mensajería del proveedor de infraestructura Big Data puede ser utilizada

para transferir datos o funciones de control a las aplicaciones que se ejecutan en las plataformas de procesamiento.

- **Visualización.** La actividad de visualización prepara los elementos de los datos procesados y los resultados de la actividad de análisis para su presentación al consumidor de datos. El objetivo de esta actividad es presentar los datos con un formato que permita expresar de manera óptima su significado y el conocimiento que aporta. Los formatos de presentación pueden ser informes basados en texto o mostrar los resultados del análisis en forma gráfica. Esta actividad interactúa con la actividad de acceso, la actividad de análisis y con el proveedor de infraestructura Big Data para que se pueda ofrecer una visualización interactiva al consumidor de datos. La visualización puede ser una implementación de una aplicación, la integración de una o más librerías o se puede utilizar plataformas especializadas en el procesamiento de visualización.
- **Acceso.** La actividad de acceso se centra en la comunicación e interacción con el consumidor de datos. Del mismo modo que la actividad de recolección, esta puede ser un servicio genérico, como un servidor web o un servidor de aplicaciones configurado para gestionar peticiones específicas del consumidor de datos. Esta actividad se comporta como una interfaz entre las actividades de visualización y análisis para responder a las solicitudes del consumidor de datos.

2.2.4 PROVEEDOR DE INFRAESTRUCTURA BIG DATA

Este rol tiene los recursos o servicios generales para ser utilizados por el proveedor de aplicaciones Big Data en la creación de una aplicación específica. Existen una variedad de componentes como recursos de procesamiento, almacenamiento y redes de datos de donde el proveedor de aplicaciones Big Data puede elegir para construir un sistema específico.

El proveedor de infraestructura Big Data consiste en una o más instancias de tres subcomponentes: plataforma de infraestructura, plataforma de datos y plataforma de procesamiento, los que detallamos a continuación.

- **Infraestructura.** Este elemento proporciona todos los recursos necesarios para albergar y ejecutar las actividades de los otros componentes. Una clasificación general de estos recursos se da de la siguiente manera:
  - **Redes:** Estos son los recursos que transfieren datos de un componente de la infraestructura a otro.
  - **Computación:** Son los procesadores y la memoria física que ejecutan y mantienen el software de los demás componentes.
  - **Almacenamiento:** Proporcionan la persistencia de los datos en un sistema Big Data.
  - **Estructura o Ambiente:** Son los recursos de la estructura física como la energía, refrigeración o seguridad que deben tenerse en cuenta cuando se realiza una implementación de una plataforma Big Data, lo cual también podría delegarse si la implementación se realiza sobre servicios de terceros alojados en la nube lo que se conoce como una infraestructura como servicio (IaaS por sus siglas en inglés).
- **Plataforma de datos.** Permite la organización y distribución lógica de los datos en combinación con los accesos asociados entre las API (Application Programming Interface) y los métodos. La organización lógica de los datos puede variar desde archivos planos delimitados hasta almacenes de datos relacionales o por columna en un entorno totalmente distribuido. Los medios de almacenamiento van desde cintas de almacenamiento, medios magnéticos, discos de estado sólido hasta las memorias de acceso aleatorio. Por estas características de almacenamiento, los métodos de acceso pueden variar desde API de acceso a archivos hasta lenguajes como el SQL (Structured Query Language).

- **Plataforma de procesamiento.** Proporcionan el software necesario para dar soporte a la implementación de aplicaciones que cumplen con las características de Big Data. Esta plataforma se centra en dar soporte a la manipulación de datos que puede ser por procesamiento por lotes (batch processing) o por streaming. Cuenta con tres fases de procesamiento: la ingesta de datos, el análisis de datos y la difusión de datos, los cuales acompañan al flujo de datos a través de la arquitectura. En el procesamiento por lotes, por streaming o la combinación de ambos se puede aplicar a las tres fases antes mencionadas. Muchos algoritmos y modelos de procesamiento se han definido con la evolución del tratamiento de datos de los que se puede resaltar dos de los más conocidos en el espacio de Big Data, MapReduce y Bulk Synchronous Parallel (BSP). La principal diferencia entre ambos es que BSP puede realizar cambios en los datos que se procesan. Se deben considerar las ventajas y desventajas de ambos al momento de su implementación, por ejemplo, BSP cuenta como desventaja el alto costo en la sincronización; mientras que MapReduce no tiene un funcionamiento óptimo cuando necesita el acceso a partes específicas del conjunto de datos ya que se tendría que volver a leer todo el conjunto de datos.

2.2.5 CONSUMIDOR DE DATOS

El consumidor de datos recibe el valor de salida del sistema Big Data. Después de que el sistema añade valor a las fuentes de datos originales, el proveedor de aplicaciones Big Data entrega ese mismo tipo de interfaces funcionales hacia el consumidor de datos. El consumidor de datos utiliza las interfaces o servicios proporcionados por el proveedor de aplicaciones Big Data para obtener acceso a la información requerida. Estas interfaces pueden incluir la presentación de datos, recuperación de datos y la representación de datos.

2.2.6 CAPA DE SEGURIDAD Y PRIVACIDAD

Los temas de seguridad y privacidad afectan a todos los otros componentes de una arquitectura Big Data. Este rol interactúa con el sistema de orquestación en cuanto a políticas, requisitos y auditoría; también con el proveedor de aplicaciones Big Data y el proveedor de datos Big Data en cuanto al desarrollo, implementación y operación de estos.

Por ejemplo, en el caso de una compañía que maneje datos críticos, es probable que se desee utilizar aplicaciones de Big Data para determinar los cambios en la demografía o los cambios en las necesidades del usuario. Estos datos acerca de los usuarios necesitan ser protegidos tanto para satisfacer los requisitos establecidos y proteger la privacidad de los mismos. Por lo que, una función clave es la de definir los niveles de accesos para quienes puedan ver los datos y en qué circunstancias se les permite hacerlo.

2.2.7 CAPA DE GESTIÓN

Las características de Big Data demandan un sistema versátil, una plataforma de gestión de software, junto con la gestión y monitorización de los recursos y su rendimiento. La gestión de Big Data implica el sistema, los datos, la seguridad y las consideraciones de privacidad, mientras se mantiene una alta calidad de los datos y una accesibilidad segura.

Esta capa de gestión abarca dos grupos generales de actividades: gestión de sistemas y gestión del ciclo de vida de Big Data. El sistema de gestión incluye actividades como el aprovisionamiento, la configuración, la gestión de paquetes, gestión de software, gestión de copias de seguridad y la gestión de recursos. La gestión del ciclo de vida de Big Data implica actividades en torno al ciclo de vida de los datos como los de recolección, preparación, análisis, visualización y acceso.

2.3 TIPOS DE ARQUITECTURAS

Debido a que las empresas disponen de un volumen de datos cada vez mayor y la necesidad de analizarlos y obtener valor de ellos lo antes posible, surge la necesidad de definir nuevas arquitecturas para cubrir casos de uso distintos a los que había hasta el momento.

Las arquitecturas más comunes en estos proyectos son principalmente dos: **Lambda y Kappa.** La principal diferencia entre ambas son los flujos de tratamiento de datos que intervienen. Un par de conceptos que serían interesantes de analizar son el procesamiento batch y el procesamiento en streaming.

- El **procesamiento de datos en modo batch**, es aquel que nos permite procesar volúmenes de datos en tiempos espaciados, por ejemplo cada 10 minutos, 1 hora o diario. Para ello el sistema dispone de lotes o batch en el que almacena toda la información que va obteniendo hasta completar un periodo. Un ejemplo de este tipo de procesamiento pueden ser las transacciones de venta del producto a lo largo de un periodo sobre el que luego realizar el procesamiento. Si el volumen de datos es elevado, este procesamiento podría demorarse varios minutos o incluso horas, y en este caso en particular es probable que quienes tomen decisiones estén dispuestos a esperar ese tiempo para poder hacerlo.
- El **procesamiento de datos en modo stream** o tiempo semireal, es aquel que necesita procesar volúmenes de datos en tiempos lo más parecido a tiempo real que se pueda. Un ejemplo típico de este tipo de procesamiento pueden ser las operaciones de inversión en bolsa en la que un instante de tiempo puede ser crucial a la hora de tomar una decisión.

2.3.1 PROCESAMIENTO BATCH

Batch hace referencia a un proceso en el que intervienen un conjunto de datos y que tiene un inicio y un fin en el tiempo. También se le conoce como procesamiento por lotes y normalmente se ejecuta sin control directo del usuario.

Por ejemplo, si tenemos un conjunto de datos muy grande con múltiples relaciones, puede llevarnos del orden de horas ejecutar las consultas que necesita el cliente, y por tanto, no se pueden ejecutar en tiempo real y necesitan de algoritmos paralelos (como por ejemplo, Map Reduce). En estos casos, los resultados se almacenan en un lugar diferente al de origen para posteriores consultas.

Otro ejemplo, si tenemos una aplicación que muestra el total de habitantes por población, en vez de realizar el cálculo sobre el conjunto completo de los datos, podemos realizar una serie de operaciones que hagan esos cálculos y los almacenan en tablas temporales (por ejemplo, mediante INSERT ... SELECT), de manera que si queremos volver a realizar la consulta sobre todos los datos, accederemos a los datos ya calculados de la tabla temporal. El problema es que este cálculo necesita actualizarse, por ejemplo, de manera diaria, y de ahí que haya que rehacer todas las tablas temporales.

2.3.2 PROCESAMIENTO STREAMING

Un procesamiento es de tipo streaming cuando está continuamente recibiendo y tratando nueva información según va llegando sin tener un fin en lo referente al apartado temporal. Este procesamiento se relaciona con el análisis en tiempo real. Para ello, se utilizan diferentes sistemas basados en el uso de colas de mensajes. Algunas de las herramientas más utilizadas son **Apache Storm, Spark Streaming, Apache Fink, Apache Kafka**.

Uno de los sistemas más populares es **Apache Storm**, que cuenta con dos tipos de nodos: nodos "**Spouts**" y nodos "**Bolts**". Los spouts convierten flujos de datos en tiempo real en flujos

de tuplas clave-valor y los emiten hacia nodos bolts que ejecutan tareas sencillas, como la lectura o escritura de una base de datos o un procesamiento simple de la tupla. Opcionalmente vuelven a emitir la tupla hacia otro nodo bolt. Cada spout y bolt es ejecutado en paralelo en múltiples ordenadores.

Los nodos Bolt emiten señales de ACK o de FAIL para notificar que la tarea ha sido o no ejecutada, haciendo así al sistema más confiable. Estos envíos se hacen a través de nodos Bolt especializados únicamente en esta tarea. A caballo entre los dos sistemas está el **Micro-batching**, que es una técnica que permite empaquetar flujos (stream) de datos entrantes en paquetes para su tratamiento por un sistema de procesamiento por lotes. Un ejemplo es **Trident**, una abstracción de alto nivel basada en Apache Storm. Trident divide los lotes en particiones, cada una orientada a ser ejecutada por un nodo Bolt.

Otro de los referentes en Big Data como motor de procesamiento de datos a gran escala es Apache Spark a través de su extensión **Spark Streaming**, un sistema de computación en clusters, de propósito general caracterizado por su alta velocidad. Apache Spark se basa en un módulo core que proporciona funcionalidad básica para planificación y gestión de tareas y de entrada y salida de datos. Define un concepto especialmente relevante, denominado RDD (en inglés "Resilient Distributed Datasets") que constituyen colecciones lógicas de datos distribuídas entre varias máquinas.

Su arquitectura está orientada al procesamiento con la memoria RAM (en inglés "in-memory") en lugar con estar orientado a trabajar con la memoria en disco duro, como hace Hadoop. Esto permite un rendimiento muy superior en algunos tipos de procesamiento, por ejemplo los que se utilizan en Machine Learning.

Figura 2.2. Spark Streaming

Spark streaming, permite el procesamiento de flujos continuos de datos a los que se les aplican funciones de alto nivel, como las funciones Map y Reduce y su resultado es almacenado en bases de datos, sistemas de ficheros distribuidos o publicados en sistemas de visualización Big Data.

2.3.3 PROCESAMIENTO MAPREDUCE

MapReduce es un modelo de procesamiento de datos, proveniente de un paradigma de la programación denominado "paradigma funcional". Este tipo de soluciones, diseñadas hace decenios y resueltas por los lenguajes de programación funcionales, son hoy especialmente relevantes ya que existen muchos problemas del mundo real que pueden ser solucionadas aplicando este modelo y específicamente muchos relacionados con Big Data.

Hasta hace pocos años no se ha dispuesto de una infraestructura de hardware y software que hiciera viable desde un punto de vista técnico y económico el aplicar este tipo de técnicas a cantidades masivas de datos. El gran tiempo de computación necesario hacía inviable la aplicación del paradigma funcional al tratamiento masivo en tiempo real de la información.

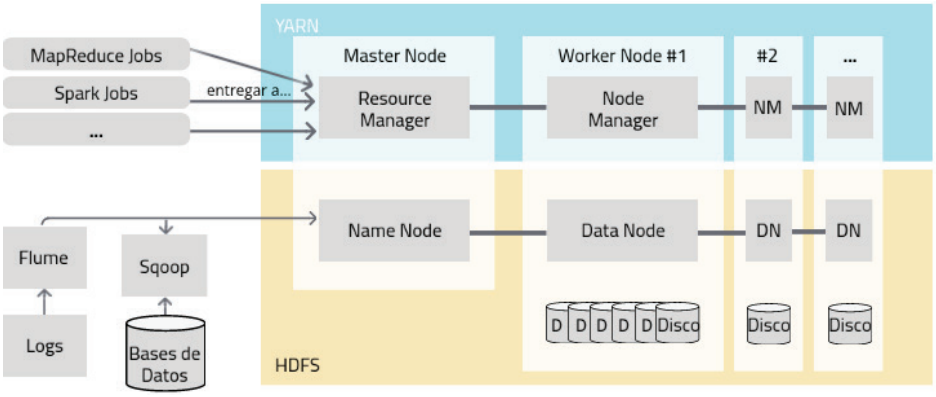

Figura 2.3. Arquitectura MapReduce

Los nuevos sistemas distribuidos si permiten este tipo de soluciones mediante paralelización en clusters de ordenadores estándar de precio reducido. Cada fragmento de trabajo en los que es dividida cada aplicación es ejecutado en un nodo del sistema distribuido.

MapReduce consiste en la unión de dos funciones de alto nivel: "**Map**" y "**Reduce**". Cada una de estas funciones de alto nivel toman como entrada una lista de pares clave-valor y su propia función, que llamaremos función-map y función-reduce.

Vamos a explicarlo con un ejemplo que usamos todos los días: las búsquedas en Google. Cuando hacemos una búsqueda en su sistema, Google nos presenta los resultados en un orden concreto. Para ello recorre con sus Bots internet buscando y contabilizando los enlaces que cada página hace a otras páginas, decidiendo así qué página es más relevante y por tanto debe ser presentada antes que otras en las páginas de resultados.

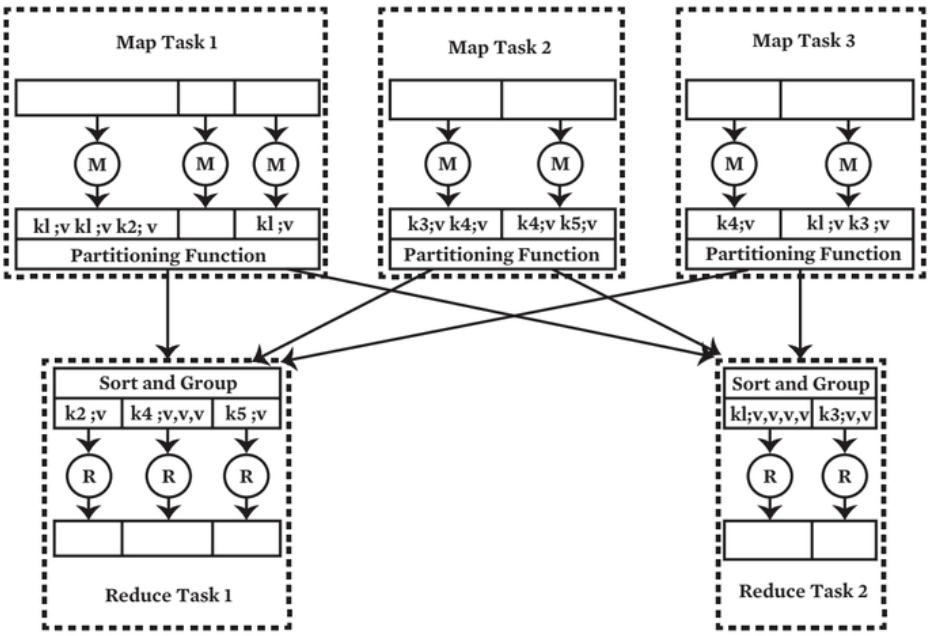

Figura 2.4. Ejecución de tareas MapReduce

En la figura anterior vemos que se han creado tres nodos Map y dos nodos Reduce. La función de Map consiste en recorrer cada página web, extraer los enlaces (en la figura los llama k1, k2, k3, k4 y k5) y les da un valor (en la figura lo llama "v", digamos para simplificar que es 1). Este sistema es muy habitual en Big Data: se le llama pares de clave-valor (en inglés "Key-Value"), de ahí el uso de "K" y de "V".

Se han generado asimismo dos nodos Reduce, cada uno de ellos recogiendo el conteo de diferentes palabras, k2-k4-k5 el de la izquierda y k1-k3 el de la derecha. La función Reduce consistirá en coger de cada nodo Map las ocurrencias de las palabras que está contando y contabilizarlas y ordenarlas. El resultado final será una lista ordenada como la siguiente:

```
(k4,3), (k1,3), (k3,2), (k5,1), (k2,1)
```

2.4 ARQUITECTURA LAMBDA

La arquitectura Lambda es una arquitectura de procesamiento genérica que posee algunas características que la han hecho ser una de las arquitecturas mayormente implementadas cuando se busca procesar información de grandes volúmenes.

Esta arquitectura combina el procesamiento de datos: «batch» y «stream», buscando las ventajas que nos ofrece cada uno de ellos. Esta arquitectura se ha desarrollado enormemente con la llegada del big data que proporciona una solución de bajo costo para problemas de procesamiento complejos.

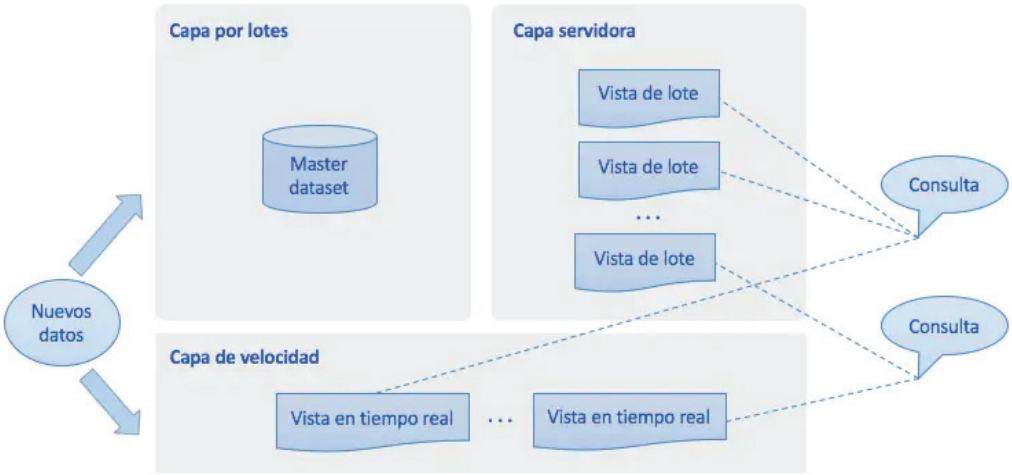

Figura 2.5. Esquema base de una arquitectura lambda

Dicha arquitectura permite ejecutar una gran cantidad de cargas de trabajo y de casos de uso para los cuales son requeridas lecturas y actualizaciones de baja latencia. El diseño de una arquitectura lambda permite su escalabilidad lineal, es decir, su escalamiento es del tipo scale-out y no scale-up, dado que el enfoque de esta arquitectura es el procesamiento de información de alta demanda usando archivos "siempre abiertos".

La arquitectura Lambda está conformada por tres capas principales que son consideradas las responsables de realizar la ejecución de las tareas más relevantes del procesamiento de datos y las que entregan los resultados de dicho procesamiento:

- **Capa de procesamiento batch**, de segmentos o lotes.
- **Capa serving**, de servidor o de consulta.
- **Capa de procesamiento streaming**.

En la arquitectura Lambda, los datos entran por duplicado, en la capa batch y en la capa streaming. A partir de aquí, se someten a dos tipos de tratamiento:

1. **Procesamiento batch**. Aborda los datos por lotes, conjuntos con un inicio y un final acotados. Ofrece outputs muy fiables, dado que toma en cuenta segmentos completos de registros; pero, a cambio, precisa de un tiempo relativamente largo (minutos u horas) para completar la operación, por lo que no se considera útil en situaciones en las que la toma de decisiones deba ser casi instantánea.
2. **Procesamiento streaming**. Proporciona información en tiempo real con vistas que se muestran directamente en la propia capa de velocidad y que se actualizan de forma constante apoyándose en los datos más recientes.

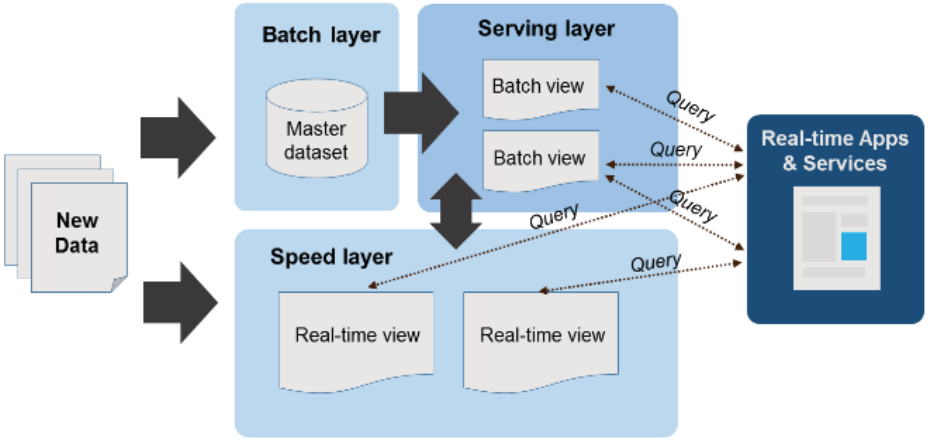

Figura 2.6. Capas de una arquitectura lambda

Otro punto a destacar de esta arquitectura es que se consigue reducir la complejidad porque evita la necesidad de escrituras aleatorias en el sistema de base de datos y esto hace innecesarios los mecanismos de bloqueo. La arquitectura lambda propone una división en tres capas entre las que podemos destacar:

- **Capa por Lotes (Batch Layer).** Esta computación se ejecuta en batch, se repite continuamente en iteraciones y conforme llegan nuevos datos no se procesan inmediatamente, sino que se encolan en el conjunto de datos maestro para agregarse a las vistas en la siguiente iteración.
- **Capa de Velocidad (Speed Layer).** El resultado de la ejecución de la capa batch no satisface el requisito de tiempo real ya que su ejecución puede llevar tiempo en la creación y propagación de las vistas. Esta capa de velocidad compensa la carencia de la capa batch y permite disponer de resultados actualizados.
- **Capa Proveedora (Serving Layer).** Es responsable de servir los datos que proceden de las dos capas anteriores: por una parte los resultados de la computación en batch y por otra los incrementos procedentes de datos nuevos. Por tanto se encarga de indexar y exponer las vistas (de sólo lectura) para que puedan ser consultadas. Las consultas se realizan bajo demanda.

Los datos que fluyen por la capa de velocidad/streaming tienen la restricción de latencia que impone la propia capa para poder procesar los datos todo lo rápido que sea posible. Normalmente, este requisito choca con la precisión de los datos. Por ejemplo, en un escenario donde tenemos dispositivos IoT donde se leen un gran número de sensores de temperatura que envían datos de telemetría, la capa de velocidad se podría utilizar para procesar una ventana temporal de los datos que entran (por ejemplo, los diez primeros segundos de cada minuto).

Los datos que fluyen por el capa de batch no están sujetos a los mismos requisitos de latencia, lo que permite una mayor precisión computacional sobre grandes conjuntos de datos, que pueden conllevar mucho tiempo de procesamiento.

Finalmente, ambos caminos convergen en las aplicaciones analíticas del cliente. Si el cliente necesita información en tiempo real aunque menos precisa, obtendrá los datos del camino rápido. Si no, lo hará a partir de los datos de la capa batch. Dicho de otro modo, el camino rápido tiene los datos de una pequeña ventana temporal, la cual se puede actualizar con datos más precisos provenientes de la capa batch. El flujo de trabajo es el siguiente:

1. La nueva información recogida por el sistema se envía tanto a la capa batch como a la capa de streaming (Speed Layer en la imagen anterior).
2. En la capa batch (Batch Layer) se gestiona la información en crudo, es decir, sin modificar. Los datos nuevos se añaden a los ya existentes. Seguidamente se hace un tratamiento mediante un proceso batch cuyo resultado serán las Batch Views, que se usarán en la capa que sirve los datos para ofrecer la información ya transformada al exterior.
3. La capa que sirve los datos (Serving Layer) indexa las Batch Views generadas en el paso anterior de forma que puedan ser consultadas con tiempos de respuesta muy bajos.
4. La capa de streaming compensa la alta latencia de las escrituras que ocurre en la serving layer y solo tiene en cuenta los datos nuevos (incrementos entre los procesos batch y el momento actual).
5. Finalmente, combinando los resultados de las Batch Views y de las vistas en tiempo real (Real-time Views), se construye la respuesta a las consultas realizadas.

Las **características** de la Arquitectura Lambda son:

- La nueva información recogida por el sistema se envía tanto a la capa de batch como a la capa de streaming, también denominada como **Speed Layer**.
- En la **capa batch (Batch Layer)** se gestiona la información en crudo, es decir, sin modificar. Los datos nuevos se añaden a los ya existentes. Seguidamente se hace un tratamiento mediante un proceso batch cuyo resultado serán las denominadas Batch Views, que se usarán en la capa que sirve los datos para ofrecer la información ya transformada al exterior.
- La capa que sirve los datos o **Serving Layer**, indexa las Batch Views generadas en el paso anterior de forma que puedan ser consultadas con baja latencia.
- La capa de streaming o Speed Layer, compensa la alta latencia de las escrituras que ocurre en la serving layer y solo tiene en cuenta los datos nuevos.
- Finalmente, la respuesta a las consultas realizadas se construye combinando los resultados de las **Batch Views** y de las **vistas en tiempo real (Real-time Views**), las cuales se han generado en el paso anterior.

La idea principal de la arquitectura Lambda es que toda la información que entra al sistema sea replicada en ambas, en la capa de velocidad (speed layer) y en la capa de segmentos (batch

layer) para que la información esté disponible para generar vistas en tiempo real en la capa de velocidad (speed layer) y vistas batch en la capa de servicios (serving layer). De esta manera, cualquier consulta realizada al sistema puede ser resuelta combinando resultados de las vistas batch (batch views) y de las vistas en tiempo real (real-time views).

La capa de segmentos (batch layer) tiene dos funciones principales:

- Administrar el dataset maestro, que es un conjunto de información en bruto (inmutable y solamente de almacenamiento).
- Llevar a cabo un pre-cómputo que genere las vistas por segmentos (batch views) que serán utilizadas en la capa de servicios (serving layer).

Por su parte, la capa de servicios (serving layer) indexa las vistas por segmento (batch views) generadas por la capa de segmentos (batch layer) para que estén disponibles para ser consultadas con baja latencia y con consultas ad-hoc.

Finalmente, la capa de velocidad (speed layer) o también conocida como capa de procesamiento entrega actualizaciones de información con alta latencia a la capa de servicios (serving layer) utilizando sólo información recientemente actualizada en la fuente de datos y creando vistas de tiempo real (real-time views) a través de algoritmos incrementales.

Un ejemplo de este tipo de arquitectura podría ser un sistema de recomendación de productos, el cual necesita extraer los productos de diferentes fuentes, procesarlos y normalizarlos, indexarlos y almacenarlas para que estén disponibles. Entre las principales **ventajas** de esta arquitectura podemos destacar:

- Los datos de entrada prevalecen intactos en una parte de almacenamiento inicial (master dataset).
- Permite que flujos de procesamiento sean rastreables y a la vez permite que se pueda hacer debug de cada etapa de manera independiente.
- Considera el problema de reprocesamiento de información una vez que la aplicación construida haya evolucionado y necesita procesar campos o variables que no eran requeridos anteriormente o simplemente cuando se quiera corregir un error.
- Combina resultados de ambos procesamiento de datos en batch y en tiempo real.

Típicamente, cada capa se puede implementar mediante las tecnologías indicadas en la siguiente tabla:

Capa	Tecnologías
Capa por lotes/Batch Layer	**Apache Spark Core** **Apache Hadoop (Map-Reduce)**
Capa de velocidad/Speed Layer	**Apache Spark Streaming** **Apache Storm** **Apache Samza** **Moa**
Capa Proveedora/Serving Layer	**Apache Hive** **ElephanDB** **Druid** **Cloudera Impala** **Apache Cassandra**

2.5 ARQUITECTURA KAPPA

Con el objetivo de mejorar la arquitectura lambda, la arquitectura Kappa trata de mejorarla eliminando el procesamiento por lotes o capa batch y la capa de velocidad (speed layer). La idea principal es administrar ambos, el procesamiento de datos y el procesamiento continuo de datos, usando un único motor de procesamiento en línea.

La idea clave es que el procesamiento por lotes también se puede llevar a cabo en la capa streaming. Y, como consecuencia, la arquitectura Kappa propone eliminar la capa de batch o de segmentos, quedando sólo con la de streaming y la de consulta, y pasando a considerar todo como un flujo de datos ininterrumpido, sin final definido, en el que aplicar las operaciones.

Partiendo de ese objetivo, la arquitectura Kappa, cuenta con dos capas: capa de procesamiento en tiempo real y capa de servicios. Se trata de una arquitectura cuyo objetivo es mejorar la arquitectura Lambda, **eliminando la capa batch dejando solamente la capa de streaming**. Esta capa, a diferencia de la de tipo batch, no tiene un comienzo ni un fin desde un punto de vista temporal y está continuamente procesando nuevos datos a medida que van llegando. Como un proceso batch se puede entender como un stream acotado, podríamos decir que **el procesamiento batch es un subconjunto del procesamiento en streaming**.

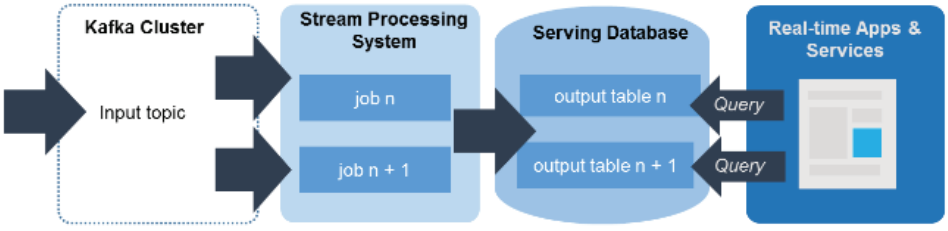

Figura 2.7. Capas de una arquitectura Kappa

Esta evolución consiste en una simplificación de la arquitectura Lambda, en la que se elimina la capa batch y todo el procesamiento se realiza en una sola capa denominada de tiempo real o Real-time Layer, dando soporte a procesamientos tanto batch como en tiempo real. El diagrama de arquitectura estaría representado por la siguiente imagen:

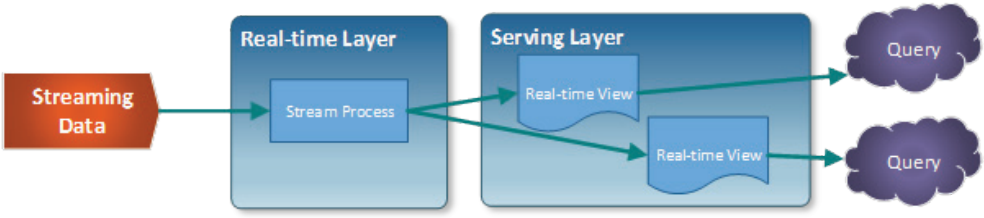

Figura 2.8. Diagrama de una arquitectura Kappa

Entre las principales **características** de esta arquitectura podemos destacar:

- **Todo es un stream**: las operaciones batch son un subconjunto de las operaciones de streaming, por lo que todo puede ser tratado como un stream.

- **Los datos de partida no se modifican**: los datos son almacenados sin ser transformados y las vistas se derivan de ellos. Un estado concreto puede ser recalculado puesto que la información de origen no se modifica.
- **Solo existe un flujo de procesamiento**: puesto que mantenemos un solo flujo, el código, el mantenimiento y la actualización del sistema se ven reducidos considerablemente.
- **Posibilidad de volver a lanzar un procesamiento**: se puede modificar un procesamiento concreto y su configuración para variar los resultados obtenidos partiendo de los mismos datos de entrada.

Como requisito previo a cumplir, se tiene que **garantizar que los eventos se leen y almacenan en el orden en el que se han generado**. De esta forma, podremos variar un procesamiento concreto partiendo de una misma versión de los datos. Entre las principales ventajas de esta arquitectura podemos destacar:

- Es una simplificación de la arquitectura Lambda, ya que se suprime el uso de la capa de batch.
- La información es almacenada utilizando un log inmutable de sólo almacenamiento, del cual se envía a almacenamientos auxiliares para la capa de serving.
- La arquitectura Kappa permite que la migración y la reorganización de la información a partir de diversas fuentes de información se ejecuten de manera eficiente proporcionando la información de manera rápida a través de la capa de streaming.
- Dada la ausencia de la capa de batch, sólo un código debe de ser actualizado en caso de necesitar mantenimiento.
- No utiliza un esquema de base de datos relacional o un almacenamiento basado en valores clave, como SQL o Cassandra, respectivamente.

Un ejemplo de aplicación utilizando esta arquitectura se compone en primer lugar de una capa de almacenamiento como **Apache Kafka**, que además de recolectar datos, sea flexible a la hora de poder cargar conjuntos de datos los cuales puedan ser reprocesados las veces que hagan falta a continuación.

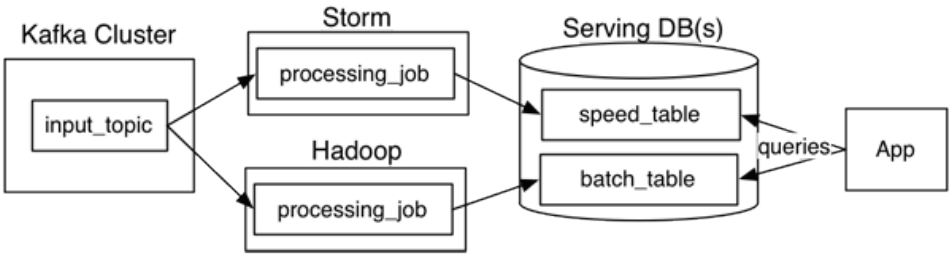

Figura 2.9. Ejemplo de aplicación de arquitectura Kappa

También disponemos de una segunda capa analítica o de procesamiento streaming como **Apache Flink o Storm**, que soporte el manejo de información asíncrona, es decir, diferenciar entre el momento en que la información fue generada (event time), momento en el que se recibe en nuestros sistemas, y momento en el que la estamos procesando.

Y por último, una capa de servicio que disponga de los resultados o de la información procesada. Aquí existe una mayor libertad a la hora de tener que elegir una tecnología o

herramienta y podrían llegar a convivir varias resolviendo cada una de ellas una determinada necesidad. En caso de estudiar, por ejemplo, las relaciones entre nuestros clientes, elegiremos una NoSQL orientada a grafos.

En esta arquitectura vemos como se eliminan los posibles puntos "débiles" de la Arquitectura Lambda y la simplifica eliminando la capa batch. Podemos afirmar que sus tres pilares principales son los siguientes:

- **Inmutabilidad de los datos.** Se garantiza que los eventos se leen y almacenan en el orden en el que se han generado. La información de origen no se modifica, los datos son almacenados sin ser transformados.
- **Solo existe un flujo de procesamiento, el de tiempo real.** Este hecho produce que el código de programación,el mantenimiento y las actualizaciones del sistema se vean reducidos considerablemente. Sin embargo, los modelos necesarios para generar las vistas precomputadas son más complejos que los empleados en la speed layer de la Arquitectura Lambda, y su entrenamiento, mucho más costoso.
- **Posibilidad de volver a lanzar un procesamiento**. Un estado concreto puede ser recalculado a posteriori partiendo de una misma versión de los datos. Cuando se necesita recomputar la información por un cambio en la lógica del negocio o en el código de la aplicación, se ejecuta la nueva función sobre el motor de stream processing que lee la información almacenada en el sistema y crea una nueva tabla con los resultados.

2.6 APACHE KAFKA

Kafka es un sistema de mensajes de tipo public-subscribe. En esencia es una cola de mensajes que permite a cualquier sistema enviar eventos, como pueden ser logs, que quedan almacenados y etiquetados, de forma que cualquier otro sistema interesado en alguno de los datos simplemente tenga que suscribirse a una etiqueta concreta.

Entre sus propiedades principales están la escalabilidad, que además al ser un sistema distribuido se puede hacer en caliente sin parar el servicio. Entre las principales **características** podemos destacar:

- Desarrollado en Scala.
- Dispone de conectores para la integración con JMS, sistema de archivos, Hadoop(HDFS), HBase, FTP, JDBC, MongoDB, Cassandra, API REST,…
- Permite la implementación de productores/consumidores en diferentes lenguajes como Java, Scala, Python, Ruby, C++
- Utiliza un protocolo propio agnóstico que va sobre http.
- Utiliza Apache Zookeeper para almacenar el estado de los nodos, que mantiene un conjunto de particiones de cada tópico.
- Diferentes grupos de consumidores pueden consumir mensajes a diferente ritmo.
- La durabilidad, ya que persiste los mensajes en disco y proporciona replicación dentro del clúster.
- La fiabilidad, mediante la replicación de datos.

En Kafka se distinguen 4 conceptos o componentes principales. Los "**topics**", que son las etiquetas que permiten definir la categoría de los mensajes que se publican. Los "**productores**", que son componentes que publican mensajes de uno o varios "topics". Los "**consumers**" son los

elementos que se suscriben a los "topics" para procesar los mensajes publicados. Y finalmente los "**brokers**" que son los servidores del clúster que gestionan la persistencia y la replicación de los mensajes.

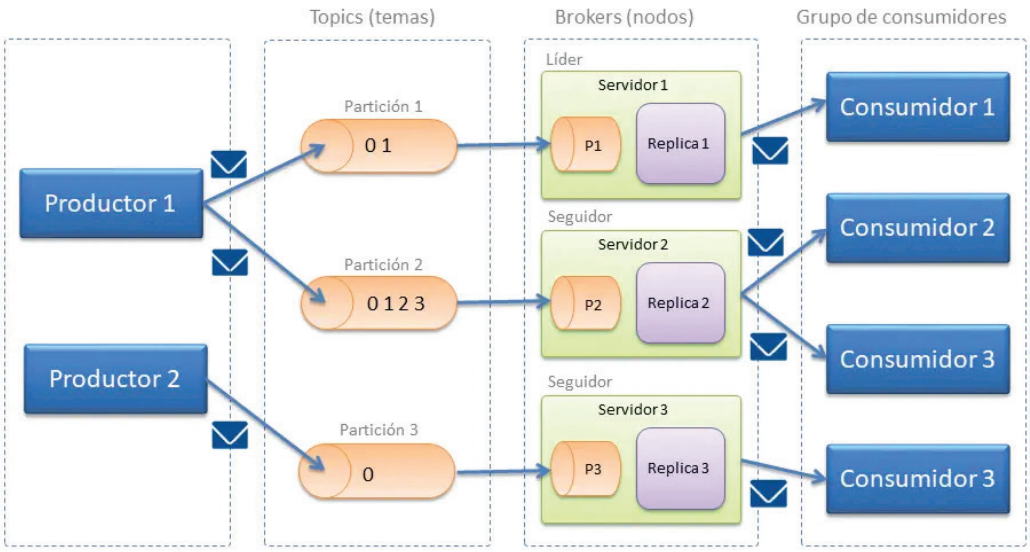

Figura 2.10. Arquitectura que representa un modelo public-subscribe

La funcionalidad de los brokers consiste en almacenar una partición con los mensajes que se envían secuencialmente de un topic, y no son responsables de controlar si esos mensajes se han consumido o no, derivando esta lógica al consumidor, que puede incluso recuperar mensajes anteriores mediante funciones de offset, dada la persistencia de los mismos. Dicha persistencia es configurable, para indicar a Kafka cuando debe dejar de guardar los mensajes.

- **Topic (tema)**: Categorías en las que clasificar los mensajes enviados a Kafka.
- **Producer (productor):** Clientes conectados responsables de publicar los mensajes. Estos mensajes son publicados sobre uno o varios topics.
- **Consumer (consumidor):** Clientes conectados suscritos a uno o varios *topics* encargados de consumir los mensajes.
- **Broker (nodos):** Nodos que forman el cluster.
- **Offset**: es el indicador que indica a cada consumidor el último elemento que ha leído. Esto hace que si se cae el sistema no se pierdan los datos.

Apache Kafka divide cada topic en particiones y cada partición es una secuencia ordenada de mensajes y cada partición es consumida por un único consumidor. A cada topic se le puede definir un número de particiones, en función del número de servidores y de conexiones que vayamos a tener. Esto aumenta considerablemente la disponibilidad.

Cada topic tiene un offset para que cada consumidor indique qué mensaje quiere que se le devuelva. A mayor número de particiones más tardará el productor (escribe mensajes) en guardar el mensaje, pero tardará menos el consumidor (lee mensajes) en recuperarlo. La idea está pensada para procesamiento en paralelo donde cada mensaje publicado en un topic se entrega a una instancia de consumidor dentro de cada grupo de consumidores suscriptores.

2.7 ARQUITECTURA POR CAPAS

Otra forma de diseñar las capas de una arquitectura big data consiste en separar las diferentes fases del dato en capas diferenciadas. La arquitectura por capas da soporte tanto al procesamiento batch como por streaming. La siguiente arquitectura consiste en 6 capas que aseguran un flujo seguro de los datos:

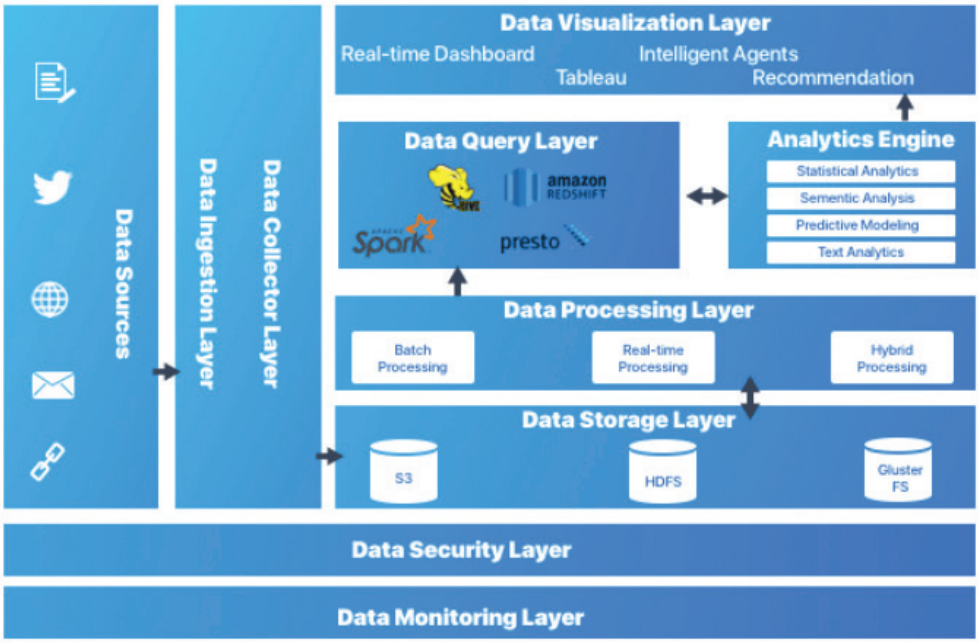

Figura 2.11. Arquitectura por capas para el procesamiento de datos

- **Capa de ingestión:** es la primera capa que recoge los datos que provienen de fuentes diversas. Los datos se categorizan y priorizan, facilitando el flujo de éstos en posteriores capas.
- **Capa de colección:** Centrada en el transporte de los datos desde la ingesta al resto del *pipeline* de datos. En esta capa los datos se deshacen para facilitar la analítica posterior.
- **Capa de procesamiento:** Esta es la capa principal. Se procesan los datos recogidos en las capas anteriores (ya sea mediante procesos *batch*, *streaming* o modelos híbridos), y se clasifican para decidir hacía qué capa se dirige.
- **Capa de almacenamiento:** Se centra en decidir dónde almacenar de forma eficiente la enorme cantidad de datos. Normalmente en un almacén de archivos distribuido, que da pie al concepto de *data lake*.
- **Capa de consulta:** capa donde se realiza el procesado analítico, centrándose en obtener valor a partir de los datos.
- **Capa de visualización:** también conocida como capa de presentación, es con la que interactúan los usuarios.

2.8 CASOS DE USO DE ARQUITECTURAS BIG DATA

A continuación, exploramos diferentes casos de uso de Big Data haciendo énfasis en la arquitectura utilizada en cada uno de estos. El objetivo es poder identificar los diferentes elementos que conforman las capas de la arquitectura base de un caso Big Data (fuente de datos, integración, almacenamiento y visualización). Durante la revisión de los casos propuestos, podríamos identificar los siguientes aspectos:

- Las fuentes de datos utilizadas y los medios mediante los cuales son recolectadas.
- Las herramientas que son utilizadas para realizar la integración de las fuentes de datos.
- La ubicación dónde almacenan el resultado de la integración de las fuentes de datos.
- Las herramientas que utilizan para analizar y visualizar el resultado del proceso de integración.

El Nasdaq Stock Market es una bolsa de valores estadounidense con sede en la ciudad de Nueva York. Ocupa el segundo lugar en la lista de bolsas de valores por capitalización bursátil de las acciones negociadas. También brindan servicios de tecnología, comercio, inteligencia y listados en todo el mundo.

En 2014, migraron al almacén de datos principal de la unidad de negocios Transaction Services U.S. de Nasdaq (que opera los intercambios de acciones y opciones de Nasdaq en EE. UU.). Gracias a la migración, Nasdaq logra capacidades de almacenamiento de datos y análisis más rápidas y útiles a la vez que ha reducido los costos en un 57% gracias al cambio a Amazon Redshift y al uso de Amazon EMR para la extracción, la transformación y la carga de datos. A continuación, puedes observar a qué capa de la arquitectura conceptual corresponden cada uno de los componentes.

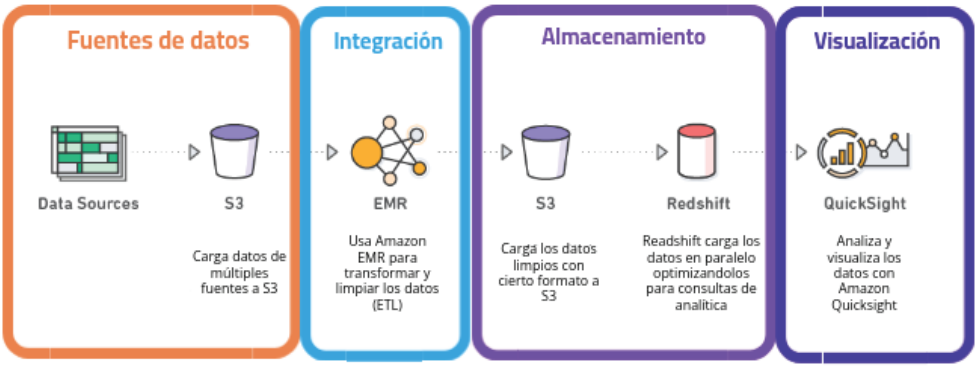

Figura 2.12. Etapas relacionadas con el tratamiento de datos

2.8.1 AUTOMÓVILES EN UN MUNDO DE STREAMING

En la actualidad, los automóviles generan información, a partir de sus sensores, que puede ser utilizada para generar valor en las compañías. Este es el objetivo principal del sistema Audi Data Collector, el cual está compuesto por diversos subsistemas para cada una de las etapas como se muestra a continuación.

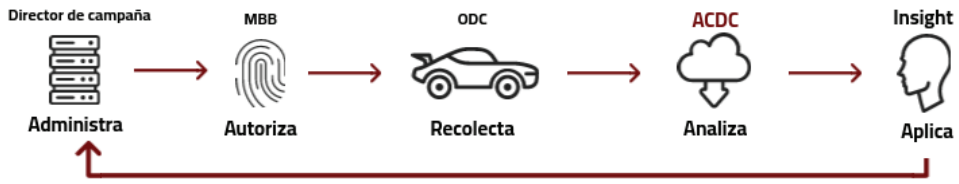

Figura 2.13. Etapas para la recolección de datos

Este sistema le permite a Audi responder a preguntas como:

- ¿Dónde encuentro un espacio libre para estacionar en mi ciudad o en cierta área?
- ¿Cuál es la ruta más óptima para llegar a mi destino teniendo en cuenta el nivel de batería y el clima?

2.8.2 CONSTRUYENDO UN SISTEMA DE LINAJE DE DATOS

El objetivo de este proyecto es construir un sistema de linaje de datos (Data Lineage) para mapear todos los artefactos de datos (incluidos los repositorios de datos en movimiento y en reposo, temas, aplicaciones, informes y paneles de Kafka, consultas de análisis interactivas y ad-hoc, aprendizaje automático y experimentación)) es una tarea muy compleja que requiere una arquitectura escalable, un diseño robusto, un sólido equipo de ingeniería y, sobre todo, una increíble colaboración interfuncional.

En la etapa de inicio del proyecto, se definió un conjunto de objetivos de diseño para ayudar a guiar la arquitectura y el trabajo de desarrollo para el linaje de datos a fin de ofrecer un sistema de linaje completo, preciso, confiable y escalable que mapee el diverso panorama de datos de Netflix. Entre los principales principios de este sistema podemos destacar:

- Garantizar la integridad de los datos.
- Permitir la integración perfecta.
- Diseñar un modelo de datos flexible.

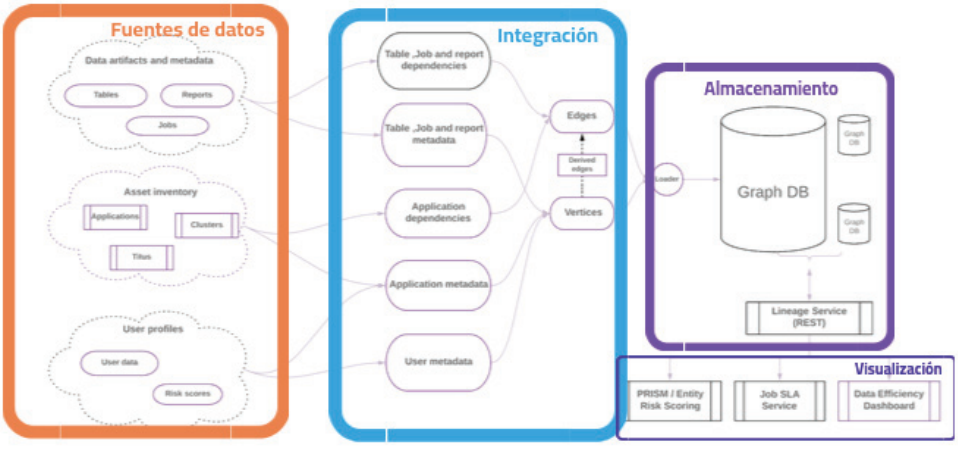

Figura 2.14. Diagrama para un sistema de linaje de datos. Fuente:*https://netflixtechblog.com/building-and-scaling-data-lineage-at-netflix-to-improve-data-infrastructure-reliability-and-1a52526a7977*

2.8.3 WOLFRAM LANGUAGE

Wolfram es el lenguaje de programación utilizado para el desarrollo de los productos de la empresa **Wolfram Research**. Es un lenguaje multi paradigma aunque lo presentan como un lenguaje basado en el conocimiento.

Wolfram Language *https://reference.wolfram.com/language/* es una manera muy visual y potente de ilustrar la funcionalidad de Machine Learning y las técnicas de Data Science. Wolfram dispone de capacidades de machine-learning en el lenguaje incluyendo aprendizaje supervisado, métodos de aprendizaje sin supervisión y de preparación y filtrado de los datos. Los datos pueden ser numéricos, textos, imágenes, etc. Entre las principales **características** podemos destacar:

- Cuenta con funciones específicas para procesamiento avanzado de textos, lingüística, redes y grafos, algoritmos de Machine Learning, imágenes, sonido, computación científica, financiera y un largo etcétera.
- Otra de las necesidades de los usuarios es diseñar modelos de los datos y deducir consecuencias a partir de dichos modelos, pudiendo obtener incluso capacidades predictivas.

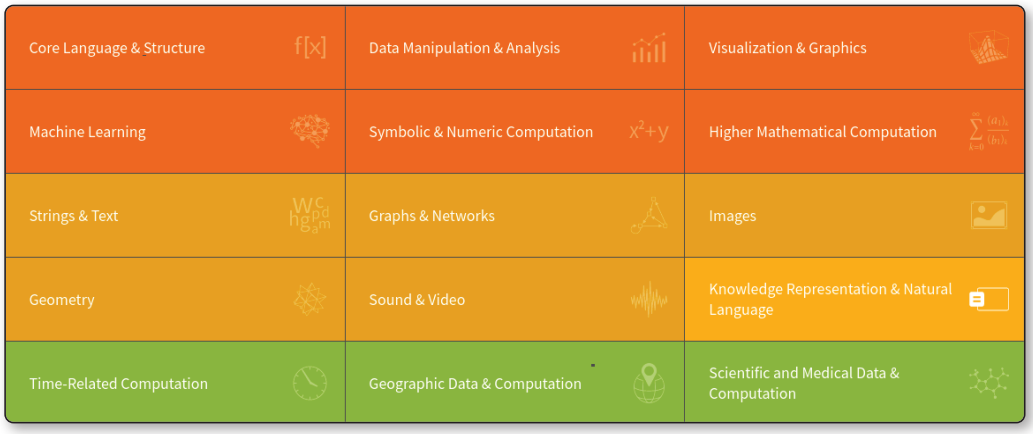

Figura 2.15. Funciones soportadas en Wolfram

De esta manera se responde de forma directa a las necesidades de los usuarios de realizar proyecciones a futuro (predicciones) y realizar clasificaciones de los datos a partir de la información disponible, incluida la información histórica almacenada.

Otra aplicación habitual es la clasificación de documentación existente obteniendo patrones de clasificación y consecuentemente usar dichos patrones en nuevos documentos a ser tratados. Por ejemplo, puede utilizarse en la búsqueda de tipos de contenidos en páginas web, como pueden ser las páginas de productos y servicios de una tienda online o de contacto en la web corporativa de una empresa.

2.9 BIG DATA LANDSCAPE

En un mundo en plena expansión y en constante evolución como es el del Big Data en esta unidad vamos a intentar exponer las líneas maestras que conforman lo que se ha venido a llamar el Universo del Big Data o "Big Data Landscape". En el sitio *http://dfkoz.com/ai-data-landscape* podemos ver las principales compañías y herramientas que la componen.

Company	Category	Sub-Category	Acquired/IPO	Exit Year	Data Driven / Hardwired Video
Amazon QuickSight	Analytics	BI Platforms			
AtScale	Analytics	BI Platforms			
Bime Analytics	Analytics	BI Platforms	Acquired by Zendesk in October 2015	2015	
Birst	Analytics	BI Platforms	Acquired by Infor in April 2017	2017	
Domo	Analytics	BI Platforms	IPO in June 2018	2018	
GoodData	Analytics	BI Platforms			
Information Builders	Analytics	BI Platforms			
Jaspersoft	Analytics	BI Platforms	Acquired by Tibco in April 2014	2014	
Kyvos Insights	Analytics	BI Platforms			
Microstrategy	Analytics	BI Platforms			

Figura 2.16. Herramientas BIG DATA organizadas por categorías

En el siguiente repositorio *https://github.com/qaware/big-data-landscape* podemos encontrar algunos diagramas del ecosistema de herramientas Big Data clasificadas por categorías. Estos diagramas nos dan una clave de la extremada importancia que ha adquirido el Big Data en los últimos años dando lugar a esta enorme cantidad de compañías que quieren vivir de explotar su tecnología.

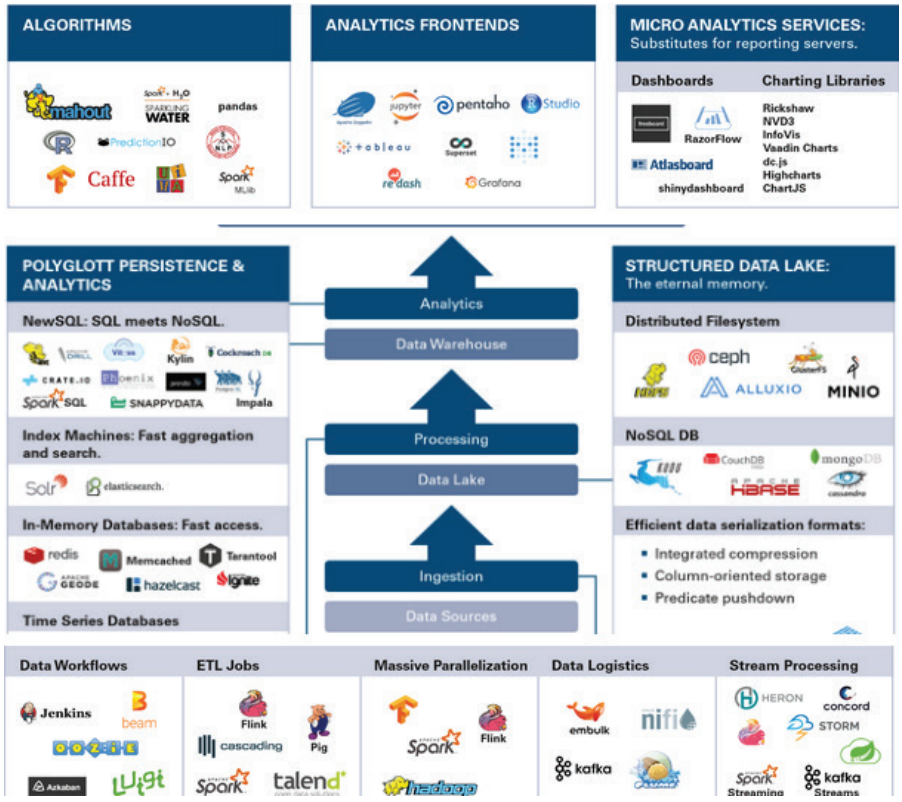

Figura 2.17. Tecnologías Big Data para diferentes casos de uso

Otra interesante referencia es *https://aplicaciones.campusbigdata.com* que proporciona un buscador y un menú con diferentes categorías desde el cual se pueden filtrar diferentes herramientas de big data.

Figura 2.18. Buscador de tecnologías Big Data

También podríamos ver una descripción de cada herramienta junto con un enlace a la página oficial:

Figura 2.19. Descripción de Apache Flink

Para superar los desafíos impuestos por la velocidad de generación, el volumen y la variedad de formatos de datos, existen múltiples tecnologías tanto de almacenamiento como de procesamiento de datos. En cuanto a almacenamiento tenemos varios caminos posibles, entre ellos el tener un sistema de archivos distribuido, un almacén de objetos (object storage), un sistema manejador de bases de datos NoSQL o uno relacional.

La decisión entre sistemas de archivos distribuidos y almacenes de objetos, dada sus similitudes, se basa en el ecosistema de herramientas a emplear y el costo de uso esperado. En este contexto, Hadoop es un nombre esperado y **Cloudera** *https://www.cloudera.com*, como

la distribución más popular a nivel empresarial del ecosistema de Hadoop. En el lado de los sistemas de almacenamiento de objetos, proveedores en la nube como Amazon, con S3 *https://aws.amazon.com/es/s3*, o Google con Cloud Storage *https://cloud.google.com/storage* son herramientas a tener en cuenta.

Para sistemas manejadores de bases de datos NoSQL la decisión se basa en los requerimientos puntuales de la aplicación, tanto funcionales como de atributos de calidad. Este entorno es extenso y los casos de usos variados, por lo que una herramienta como **QuABaseBD** *http://quabase.sei.cmu.edu* de la universidad Carnegie Mellon, que permite consultar las herramientas de acuerdo a los atributos de calidad, es especialmente útil.

En procesamiento, aunque MapReduce sigue siendo relevante, con implementaciones como la de Hadoop en Cloudera para ambientes on-premise o EMR de AWS para Cloud *https://aws.amazon.com/es/emr/*, Spark es cada vez más fuerte. Spark está incluido en muchas de las distribuciones de Hadoop, pero no requiere a Hadoop para ser ejecutado por lo que puede ser desplegado en entornos diversos.

En la nube el mayor proveedor de Spark es Databricks *https://databricks.com*, con su estrategia multi-nube, completamente administrada. Una estrategia alternativa es la de Data Mechanics *https://www.datamechanics.co*, que fueron pioneros en el despliegue de **Spark** sobre Kubernetes en la nube pública.

En cuanto a procesamiento de flujos de datos, Kafka *https://kafka.apache.org* es el sistema distribuido de flujo de eventos por excelencia, pero no es el único. Herramientas on premise de colas de mensajes, como ActiveMQ *https://activemq.apache.org* o RabbitMQ *https://www.rabbitmq.com*, pueden cubrir parte de las funcionalidades. En el entorno de la nube Confluent *https://www.confluent.io* ofrece un servicio de Kakfa completamente administrado y existen competidores como Amazon Kinesis *https://aws.amazon.com/es/kinesis/* que incluyen capacidades de procesamiento.

Para el procesamiento de flujos de datos Spark Structured Streaming *https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html* , sobre alguna de las plataformas mencionadas anteriormente, es una opción. Otros frameworks como Apache Flink *https://flink.apache.org*, o Apache Storm *https://storm.apache.org* tienen funcionalidades similares aunque su forma de ver los flujos de datos sea diferente.

2.10 HERRAMIENTA PARA EL ANÁLISIS DE DATOS MASIVOS

En los últimos años, ha surgido un gran abanico de herramientas de analítica de datos escalables asociadas a algunas de las plataformas comentadas anteriormente, con el objetivo de dar soporte al proceso de análisis de datos. A continuación, describimos brevemente algunas:

- **Apache Mahout** *https://mahout.apache.org*: Esta biblioteca ofrece implementaciones basadas en Hadoop MapReduce para varias tareas de analítica de datos como el agrupamiento, la clasificación o el filtrado colaborativo.
- **Spark MLlib** *https://spark.apache.org/mllib*: Nacida junto al proyecto de Spark, es una biblioteca de aprendizaje automático que contiene varias utilidades estadísticas y algoritmos de aprendizaje. Esta biblioteca contiene algoritmos que dan soporte a tareas del proceso de extracción del conocimiento como clasificación, optimización, regresión, agrupamiento y preprocesamiento.
- **FlinkML** *https://github.com/apache/flink-ml*: Es la biblioteca para análisis distribuido de datos de Flink. FlinkML incluye algoritmos escalables para tareas como la clasificación, el agrupamiento, el preprocesamiento de datos y la recomendación. Aunque está lejos

de ofrecer la variedad de otras bibliotecas como MLlib, ofrece algunas técnicas que se proponen mejorar a las de MLlib, como por ejemplo la implementación del algoritmo de Support Vector Machines (SVM).

- **H2O** *https://h2o.ai*: Es una plataforma de código abierto para análisis Big Data. H2O destaca por su aproximación al deep learning, y por sus implementaciones iterativas. Estas últimas permiten que el usuario decida si obtener la solución más óptima u obtener una solución aproximada. H2O puede ser ejecutada en sistemas tradicionales (Windows, Linux, etc.), así como en plataformas Big Data.

2.11 CONCLUSIONES

En este capítulo se ha definido el modelo conceptual de una arquitectura Big Data, indicando los diversos elementos o capas que la conforman, además de especificar el comportamiento y la relación de cada uno de ellos. Esto nos permitirá definir una arquitectura teniendo en consideración las diferentes características de cada elemento y cómo se deben organizar en una arquitectura específica. Y de esta manera, poder seleccionar y clasificar las herramientas que permitirán la implementación del sistema Big Data.