Fundamentos de *big data*

Habilitando la explotación de datos complejos

Josep Curto

PID_00242650

Universitat Oberta de Catalunya

Índice

Introducción

En los últimos años las empresas se han embarcado en un proceso de transformación digital de profundo calado dentro del marco de lo que se conoce como la cuarta revolución industrial, que está dando paso a una nueva manera de organizar los medios de producción. Las empresas se están transformando en «fabricas inteligentes» capaces de una mayor adaptabilidad a las necesidades y a los procesos de producción, así como de una asignación más eficaz de los recursos, lo que abre la vía a una nueva revolución industrial. No solo se trata de la digitalización de los procesos de negocio, sino también del uso del dato y las tecnologías de la información (TI) para la optimización y automatización de dichos procesos. Las TI han pasado de estar en la periferia de la organización a estar en el centro y erigirse en uno de sus pilares. Esta progresiva transformación de base tecnológica se ha combinado con otros aspectos, como el advenimiento de las redes sociales, la democratización de internet o el despliegue de la internet de las cosas.

Lectura Complementaria

K. Schwab (2016). *The Fourth Industrial Revolution*. Davos: World Economic Forum

El resultado de esta tormenta perfecta en la que se hallan todas las organizaciones es una explosión del dato en volumen, velocidad y variedad. Y de modo natural ha crecido la complejidad para capturar, procesar, almacenar, analizar y visualizar los datos.

Internet de las cosas

Internet de las cosas hace referencia a la interconexión digital de objetos cotidianos con internet. Nos referiremos a ella por su acrónimo en inglés IoT, *Internet of Things*.

Como resultado, han aparecido múltiples métodos, técnicas y tecnologías que buscan ayudar a las organizaciones a tomar mejores decisiones a partir de los datos y a extraer valor de estos. Estos métodos, técnicas y tecnologías para la captura, el procesamiento, el almacenamiento, la gestión y el análisis se han ido progresivamente estructurando en diferentes estrategias que conocemos como ***big data***.

Aunque este concepto ya lleva varios años en el mercado y existen múltiples casos de uso conocidos, las organizaciones siguen teniendo problemas para conocer el impacto y valor de *big data*, y sobre todo para poner en marcha estos sistemas de información. Existen todavía múltiples preguntas:

- ¿Qué es *big data*?
- ¿Qué significa para mi organización?
- ¿Cuándo es relevante?
- ¿Está preparada mi organización?
- ¿Cómo desplegar con éxito este tipo de iniciativas?
- ¿Qué barreras presentan este tipo de proyectos?
- ¿Qué tecnologías existen dentro de *big data*?
- ¿Cómo empiezo un proyecto?

Respondiendo a las anteriores preguntas, el presente material busca capacitar a profesionales en el contexto del análisis de la información con el objetivo de desarrollar estrategias de negocio que incluyan *big data* en el seno de su propia organización. Y en consecuencia, poder detectar casos de uso y problemáticas en la propia organización que necesiten este tipo de enfoque.

Objetivos

Este material didáctico está dirigido a:

- Desarrolladores y consultores que quieren conocer *big data*.
- Desarrolladores y consultores que quieren ayudar al desarrollo de estrategias de negocio que incluyan *big data*.
- Gestores que están interesados en la transformación digital de su organización y en la inclusión de *big data* como uno de sus pilares fundamentales.

y tiene los siguientes objetivos:

1. Entender el concepto de *big data*, las situaciones en las que es necesario desplegar una solución de este tipo y las ventajas que proporciona.

2. Contextualizar qué es necesario tener em una estrategia de negocio que incluya *big data*.

3. Conocer qué significan las fases de madurez de una estrategia de *big data*.

4. Enumerar y dar a conocer las tecnologías que engloba *big data*.

5. Dar a conocer casos de uso y ejemplos.

Si bien la obra es autocontenida en la medida de lo posible, los conocimientos previos necesarios son:

- Conocimientos sobre básicos sobre *business intelligence* y *business analytics*.
- Conocimientos sobre estrategia y gestión de las tecnologías de la información (TI).

Se introducirán los conceptos necesarios para el seguimiento de este material.

1. El nuevo contexto de negocio

El uso de datos para tomar mejores decisiones no es nuevo. De hecho, desde hace tiempo las organizaciones se han estado apalancando en estrategias como la inteligencia de negocio y/o la analítica de negocio para ello. Pero una serie de condiciones en el mercado han propiciado que sea necesaria una nueva estrategia para el análisis de datos: ***big data***.

Es este primer apartado nos centraremos en comprender cuáles son estas nuevas condiciones del mercado, qué ha cambiado de la naturaleza del dato y, por último, discutiremos por qué los sistemas anteriores no son suficiente.

1.1. Qué ha cambiado desde el punto de vista de negocio

En las últimas décadas, las tecnologías de la información poco a poco han ido asumiendo mayor relevancia en las organizaciones. Por un lado, se han transformado en un componente básico para las operaciones automatizando parte o incluso todo el proceso y, por otro, proporcionan soporte a las diferentes necesidades departamentales (desde finanzas hasta marketing).

Esta ha sido una progresiva transformación digital y aún muchas empresas se encuentran en este proceso de profundo calado. En los últimos años, la transición se ha acelerado por diversos factores, como la democratización de internet, el advenimiento de las redes sociales, la emergencia de los dispositivos inteligentes y/o el despliegue de la internet de las cosas. Y como resultado las organizaciones se encuentran en un periodo de competitividad y evolución disruptiva basada en TI y fundamentada en cuatro factores principalmente: el *cloud*, lo social, la movilidad y la analítica. Expliquemos estos factores tecnológicos:

- ***Cloud***: hace referencia a tecnologías que permiten consumir recursos TI (desde almacenamiento hasta un CRM) gestionados por terceros y cuyo uso frecuentemente se comparte.

- **Social**: hace referencia a las tecnologías que permiten facilitar las interacciones sociales.

- **Movilidad**: hace referencia a las tecnologías que habilitan el acceso a información e interacciones con independencia de la localización.

- **Analítica**: hace referencia a aquellas tecnologías que maximizan la utilidad del dato.

CRM

CRM es el acrónimo de *customer relationship management*, que se refiere a la gestión de la relación con clientes.

Estos cuatro factores provocan que los modelos de negocio sean diferentes. *Cloud* permite que tengamos flexibilidad en la implementación, en el despliegue y en la escalabilidad; social redefine el modo como interactuamos con clientes, empleados y proveedores; móvil amplía los canales de interacción y desdibuja el perímetro de lo que conocemos como empresa; y, por último, la analítica significa que ahora no solo podemos conocer lo que sucede en la organización, sino también incrementar y automatizar la inteligencia en ella. En definitiva, es una transformación de la relación entre personas, negocio y tecnología.

Como resultado hemos asistido a la explosión de nuevas formas de acercarse al mercado y generar valor para el cliente y la organización. Por ejemplo, sabemos de empresas que han conseguido crear sistemas de recomendación para sus clientes, como Amazon, diseñar productos basados en preferencias, como Netflix, o identificar el riesgo crediticio basado en fuentes tan dispares de información como las redes sociales o las compras en eBay y Amazon, como Kreditech. Aunque compañías como Facebook, Google o Netflix acaparan la atención por sus avances en el uso de las tecnologías de datos, la realidad es que estamos viviendo una revolución de amplio espectro y muchas otras empresas ya están apostando por la implementación de este tipo de proyectos.

De hecho, es posible encontrar ejemplos en múltiples sectores; estas aplicaciones tienen múltiples formas y colores, y frecuentemente están profundamente verticalizadas. Por ejemplo, en el contexto de los *massively multiplayer online game* (MMOG) empresas como Jagex ya monitorizan las transacciones de micropagos y el funcionamiento de los sistemas que soportan las operaciones usando tecnologías de *big data*. En el sector del deporte, equipos como el FC Barcelona analizan grandes cantidades de datos en diferentes formatos (vídeos, estadísticas, datos geolocalizados, etc.) para comprender mejor el rendimiento propio como equipo y de forma individual, así como el de los equipos contrarios, y diseñar consecuentemente estrategias más eficientes para ganar. En el sector de la agricultura, *big data* permite mejorar la eficiencia de los sistemas de riego al ser la pieza clave para integrar y analizar datos de estaciones meteorológicas, informes de plagas y enfermedades, sensores en plantas, bocas de riego y suelo de parcelas, y sistemas más tradicionales (por ejemplo, el ERP) como en el caso de la bodega Luna Beberide.

ERP

ERP es el acrónimo de *enterprise resource planning*, que hace referencia a la gestión de los recursos de una organización.

Pero como ocurre cada vez que aparece una nueva tecnología innovadora y de vanguardia que tiene el potencial de transformar profundamente la sociedad, no resulta sencillo llevar a buen puerto la implementación. Y una primera pregunta surge: **¿en qué medida ha cambiado el dato?**

1.2. La naturaleza del dato

Tal y como hemos comentado, estamos viviendo una explosión en la complejidad del dato. Para entender esta complejidad, es necesario hablar sobre la naturaleza del dato, hacer un inciso sobre qué entendemos por las magnitu-

des físicas del dato y entrar en detalle en dos puntos cada vez más relevantes: dónde se encuentran los datos importantes para una organización y el papel crucial de los metadatos.

1.2.1. Las magnitudes físicas del dato

Hay tres magnitudes físicas del dato:

- **Volumen**. Cuando hablamos de volumen, hacemos referencia al tamaño del conjunto de los datos creado diariamente. En apenas una década, las organizaciones han pasado de trabajar con terabytes a tener de lidiar con petabytes o magnitudes superiores.

- **Velocidad**. Cuando hablamos de velocidad, nos referimos tanto al procesamiento de datos como a su latencia. El primero hace referencia a la cantidad de datos en movimiento (medida en términos de gigabytes o terabytes por segundo). El segundo se refiere a la diferencia entre la ingestión de datos y el análisis de estos (medido en milisegundos). Esto se traduce en tratar con datos mediante procesos *batch* pero también en tiempo real y/o *streaming*.

- **Variedad**. Cuando hablamos de variedad, hacemos referencia tanto a la cantidad de fuentes diferentes que combinar como a la heterogeneidad del dato (siendo estos estructurados, semiestructurados o no estructurados).

Latencia

Cuando hablamos de latencia hacemos referencia a la suma de retardos temporales en la captura, el almacenamiento, el procesamiento y el análisis del dato.

Byte

Cuando hablamos de byte hacemos referencia a una unidad de medida de información digital. Hablaremos de múltiplos de bytes:

gigabyte (GB) 10^{9} bytes

terabyte (TB) 10^{12} bytes

petabyte (PB) 10^{15} bytes

exabyte (EB) 10^{18} bytes

zettabyte (ZB) 10^{21} bytes

yottabyte (YB) 10^{24} bytes

Más allá de las magnitudes físicas es posible encontrar otras características, como:

- **Veracidad**, que hace referencia a la incertidumbre en el dato producto de su baja calidad, la ambigüedad en su definición o simplificaciones en su modelización.

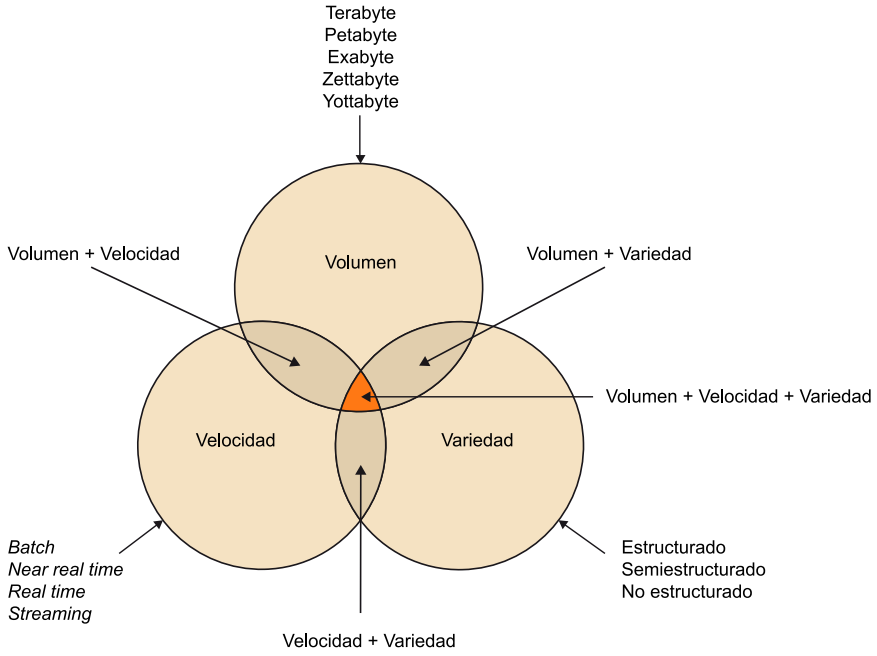
- **Variabilidad**, que se refiere a que los flujos de datos pueden tener comportamientos erráticos o inconsistentes en ciertos periodos.

- **Vinculación**, que denota la dificultad de relacionar diferentes y dispares fuentes de datos.

Cabe comentar que no siempre nos encontramos con estas características y que dependen de la naturaleza del problema que se pretende resolver. Por ello solo se habla de las tres primeras.

Diferentes problemáticas de negocio tendrán asociadas diferentes combinaciones de las magnitudes físicas, como se ilustra en la figura 1.

Figura 1. Magnitudes físicas del dato



Fuente: Josep Curto, adaptado de SmartDataCollective.

Hay un último punto que comentar con respecto la naturaleza del dato: el **valor**. Aunque hayamos discutido cómo podemos entender la complejidad del dato, lo importante no es si una organización es capaz de gestionar el dato en reposo, en movimiento y/o en sus múltiples formas y fuentes. Lo más relevante es cómo una organización es capaz de generar valor a partir del dato y qué impacto tiene para el negocio y para los clientes.

Este valor puede tomar diferentes formas:

1) Toma de decisiones: el uso del dato nos permite tomar mejores y/o más rápidas decisiones, lo que se traduce en que la organización es más competitiva en su respectivo mercado. Es decir, somos capaces de tomar decisiones informadas.

2) Ingresos: el uso del dato permite mejorar los ingresos en líneas de negocio o habilita la creación de nuevas.

3) Costes: el uso del dato permite optimizar nuestros procesos de negocio a nivel de sistemas, procesos, clientes, empleados o proveedores, lo que se traduce en que podemos hacer más con menos.

1.2.2. ¿Dónde se encuentran los datos relevantes para el negocio?

Hemos hablado en el subapartado 1.2.1 sobre el aspecto más importante del dato: su valor. O lo que es lo mismo, que el dato sea relevante para el negocio. Ante el nuevo contexto, las organizaciones necesitan trabajar con el dato dejando atrás la noción de que la información de valor se encuentra tan solo en el seno de la organización. Este hecho obliga a pensar no solo en las magnitudes del dato, sino en el origen de partida de estos. Por lo tanto, debemos hablar de los siguientes datos:

- **Datos internos**: hace referencia a datos que pertenecen a la organización. Dentro de los datos internos tenemos aquellos que ya existen o se crean en los propios sistemas de información de la organización (como pueden ser el ERP y/o el CRM), o que se están capturando y almacenando mediante mecanismos automáticos a través de diferentes estrategias, como el *crowdsourcing*, sensores y/o dispositivos de monitorización (como un podómetro con localización geográfica).

- **Datos externos**: son datos de terceros y que deben ser conseguidos por la organización. Estas fuentes de datos pueden estar disponibles para su compra o ser de acceso libre. Los datos de libre acceso pueden ser a su vez de tres tipos: datos capturados mediante técnicas de *crowdsourcing*, datos de redes sociales (como pueden ser Facebook, Twitter o LinkedIn) y *open data*.

Crowdsourcing

Cuando hablamos de *crowdsourcing* hacemos referencia al proceso de obtener servicios, ideas, contenido, etc., a través de la participación de una gran masa de personas.

Open data

Cuando hablamos de *open data* hacemos referencia a conjuntos de datos considerados que son un bien común y que, por ello, son gratuitos, accesibles y bien estructurados para su descarga y análisis. Las tipologías son múltiples: de transporte, financieros, meteorológicos, estadísticos, científicos, culturales y geolocalizados.

Sea cual sea el origen del dato, los consumidores en cualquiera de las formas de valor tienen expectativas que han de ser cubiertas. Estamos hablando de que para tomar una decisión el dato debe estar disponible (ha sido capturado y almacenado), accesible (existe un mecanismo para el consumo), de calidad (se ha validado que tiene el nivel de calidad suficiente), en el momento adecuado (se han tenido en cuenta las necesidades temporales de negocio para su disponibilidad y accesibilidad), securizado (lo que significa que está protegido y solo pueden acceder a él aquellos que tienen permisos) y transformado en información (lo que significa que quien consume el dato no necesita transformar el dato en información).

1.2.3. Más allá del valor del dato

Es necesario hablar de una última potencial fuente de datos, que, aunque podría incluirse dentro de la categoría de datos internos, no suele contemplarse dentro de las organizaciones. Estamos hablando del metadato y su valor asociado. Primero debemos definir qué es el metadato.

> «Se entiende por metadatos a datos estructurados y codificados que describen características de un objeto, dato o proceso de negocio».
>
> J. Conesa; J. Curto (2012). *Introducción al Business Intelligence*. Barcelona: Editorial UOC.

Es decir, no es suficiente con generar valor a partir del dato, sino también de los metadatos vinculados a dicho dato. Podemos hablar de tres grandes categorías de metadatos:

- **Técnicos**: que describen los aspectos técnicos vinculados al dato, como pueden ser las magnitudes del dato o, por ejemplo, los derechos de propiedad.

- **Operacionales**: que hacen referencia a los procesos de captura, transformación, almacenaje, análisis y visualización del dato.

- **Atributos**: que se refieren a los atributos que enriquecen la información sobre el dato. Por ejemplo, en una fotografía encontramos aspectos como el dispositivo con el que se realizó.

El metadato abre la puerta a una nueva gama de análisis del valor y, sobre todo, a comprender de una manera mucho más profunda lo que sucede en una organización. Información que no siempre es relevante para el usuario final, pero sí de suma importancia para el sistema que gestiona el dato.

Tras discutir la naturaleza del dato, otra pregunta natural surge: **¿por qué necesitamos una nueva tecnología para analizar el dato?**

1.3. Las limitaciones del *data warehouse*

Tradicionalmente las organizaciones han abordado su necesidad de analizar datos y generar valor a través de dos sistemas interconectadatos: el *data warehouse* –o almacén de datos– y la inteligencia de negocio –o *business intelligence* (BI). Debemos comprender primero qué son estos conceptos. Definimos qué es un *data warehouse*:

> «Se entiende por ***data warehouse*** el repositorio de datos que proporciona una visión global, común e integrada de los datos de la organización, independiente de cómo se vayan a utilizar posteriormente por los consumidores o usuarios, con las propiedades siguientes: estable, coherente, fiable y con información histórica».
>
> J. Conesa; J. Curto (2012). *Introducción al Business Intelligence*. Barcelona: Editorial UOC.

Ahora podemos definir qué es la inteligencia de negocio:

> «Se entiende por ***business intelligence*** el conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización».
>
> J. Conesa; J. Curto (2012). *Introducción al Business Intelligence*. Barcelona: Editorial UOC.

Es fácil deducir que el *data warehouse* ha sido el componente principal para el almacenamiento de datos y el BI lo ha sido para su explotación.

Sin embargo, a medida que las organizaciones han ido progresando en su transformación digital, la complejidad del dato se ha ido incrementado y nuevas necesidades han emergido. Estas son algunas de ellas:

- La toma decisiones necesita integrar datos estructurados, semiestructurados o no estructurados.
- La toma de decisiones necesita trabajar con estructuras de datos que no son persistentes en el tiempo.
- La toma de decisiones necesita considerar toda la información asociada a un proceso de negocio, lo que se traduce, para algunos de ellos, en grandes cantidades de información, no procesables de modo eficiente.
- La toma de decisiones debe realizarse en tiempo real acelerando la captura y el consumo del dato.
- La toma de decisiones debe fundamentarse en el uso de aplicaciones analíticas donde el metadato del proceso desempeña un papel fundamental en la comprensión y el descubrimiento de lo sucedido.
- El dato se reutilizará para diferentes análisis y, por ello, se necesita guardar en bruto o aplicando el mínimo de transformaciones posible.

Una manera de entender este tipo de escenarios es comparar los casos de uso del almacén de datos respecto a los nuevos casos de uso, tal y como se recoge en la tabla 1.

Tabla 1. *Data warehouse* frente a nuevos escenarios de uso del dato

Factor	*Data warehouse*	Nuevo escenarios de uso del dato
Fuentes de datos	Sistemas corporativos y transaccionales	Fuentes no tradicionales, como sensores, *logs*, vídeos, etc.
Volumen	Hasta 100 Terabytes	A partir de 100 Terabytes
Velocidad	*Batch* o procesos que no requieren respuesta inmediata	Respuesta inmediata
Variedad	Principalmente estructurada	De todo tipo

Cuadro de mando

Cuando hablamos de cuadro de mando hacemos referencia al sistema que informa de la evolución de los parámetros fundamentales de negocio de una organización o de un área de esta.

Factor	*Data warehouse*	Nuevo escenarios de uso del dato
Veracidad	Organizada y de calidad	De calidad variable
Valor	BI y analítica	*Machine learning*, *deep learning* y anteriores
Objetivo	Toma de decisiones	Múltiple, pero destaca la creación de productos y servicios de datos

La gran mayoría de las implementaciones de *data warehouse* han sido creadas de forma optimizada para la generación de informes, cuadros de mandos y el análisis OLAP. Escenarios enfocados al análisis de rendimiento pasado de una organización y que deben estar fundamentados en información de calidad y con modelo que representa procesos de negocio y perspectivas de análisis.

Los nuevos escenarios no han sido tratados por el almacén de datos e incluso no forman parte de sus capacidades y, por lo tanto, como respuesta a dicha necesidad ha emergido una nueva generación de tecnologías y enfoques que amplía las capacidades de nuestra organización a nuevos casos de uso.

OLAP

Cuando hablamos de OLAP, o proceso analítico en línea, nos referimos al método para organizar y consultar datos sobre una estructura multidimensional. A diferencia de las bases de datos relacionales, todas las potenciales consultas están calculadas de antemano, lo que proporciona una mayor agilidad y flexibilidad al usuario de negocio.

2. ¿Qué es *big data*?

En 2009, IDC estimó el tamaño de la información digital generada y guardada, a la que llamó el universo digital, en 0,8 Zettabytes (ZB) y predijo que para el año 2020 se llegarían a los 35 ZB. Posteriores estudios de la misma compañía han revisado la cifra al alza para dicho año y la han ajustado a 45 ZB (8 ZB para el año 2015). Esta revisión en las predicciones ilustra la aceleración fruto de la aparición de cada vez más fuentes que producen y consumen datos; de una mayor incorporación de usuarios a internet; del despliegue de una mayor cantidad de dispositivos inteligentes, y del continuo desarrollo de soluciones y servicios digitales.

Lectura complementaria

J. Gantz; D. Riensel (2009). *As the Economy Contracts, the Digital Universe Expands*. Nueva York: IDC.

Esta explosión de datos está caracterizada por un crecimiento en las magnitudes físicas del dato: volumen, variedad y velocidad. Se crea un mayor volumen de datos, provenientes de una mayor variedad de fuentes, representados en múltiples formatos y que se deben capturar y consumir a una mayor velocidad. Este nuevo paradigma de los datos se conoce frecuentemente ***big data***, si bien el nombre produce confusión teniendo en cuenta su referencia a solo una de las magnitudes (volumen). En esencia, estamos hablando de una explosión en la complejidad del dato.

2.1. Definición de *big data*

Se considera que *big data* es un concepto novel, al existir múltiples definiciones de él. Es necesario comentar que podemos encontrar referencias a la problemática del dato en 2001 cuando Doug Layney apuntó que el crecimiento de los datos en volumen, variedad y velocidad iba a propiciar la necesidad de invertir en nuevas tecnologías que permitieran capturar, extraer, procesar, guardar y analizar los datos en la nueva era. Pero los orígenes del término pueden encontrarse incluso antes, en la década de los noventa, en las conversaciones dentro de la comunidad de Silicon Graphics dirigidas por el científico John Mashley en las que se analizaba las principales tendencias de futuro.

Big data

Existen múltiples definiciones de *big data* tal y como se argumenta en el siguiente artículo académico: «Undefined By Data: A Survey of Big Data Definitions». Fuente: http://arxiv.org/pdf/1309.5821v1.pdf, y como también ha argumentado Timo Elliot, evangelista de SAP, en su blog: http://timoelliott.com/blog/2013/07/7-definitions-of-big-data-you-should-know-about.html.

El hecho de que existan múltiples definiciones complica su comprensión y la identificación de escenarios dentro de la propia organización. La gran mayoría de ellas incluyen lo que se conoce como las 3 V del *big data* que hemos comentado en la anterior sección: volumen, velocidad y variedad, que son magnitudes físicas del dato. No obstante, podemos encontrar otras definiciones que incluyen algunas más, como por ejemplo la veracidad.

Doug Layney

Doug Layney es un analista que en el año 2001 pertenecía a MetaGroup y que actualmente trabaja en Gartner.

Por tanto, en aras de tener un enfoque pragmático, vamos a usar la siguiente definición de *big data*.

Se entiende por ***big data*** el conjunto de estrategias, tecnologías y sistemas para el almacenamiento, procesamiento, análisis y visualización de conjuntos de datos complejos.

Es necesario recordar que cuando hablamos de almacenamiento hablamos de los soportes físicos y de software que permiten guardar el dato en estructuras que representan su complejidad; que cuando hablamos de procesamiento nos estamos refiriendo a aquellas operaciones que permiten la ingestión, la transformación y la distribución del dato para adecuarlo para el consumo; que cuando hablamos de análisis hacemos referencia a las técnicas aplicadas para generar valor, y que cuando hablamos de visualización nos referimos a los mecanismos de consumo de información.

2.2. Tipos de *big data*

La definición de *big data* enmascara en cierta medida las complejidades de lo que supone trabajar con datos extremos en términos de su volumen, velocidad y variedad. Ya hemos introducido en el subapartado 1.2 en qué consiste la nueva naturaleza del dato.

En este sentido, y con el objetivo de mejorar la comprensión de la definición de *big data*, es necesario hablar de las diferentes tipologías de *big data* existentes.

2.2.1. Clasificación de NIST

De acuerdo con el NIST, y en particular dentro de su grupo de trabajo de *big data*, una forma de categorizar *big data* es mediante las necesidades de negocio:

- El modelo de negocio no se puede representar mediante una estructura de datos relacional (es decir, mediante una base de datos relacional).
- El modelo de negocio necesita ser escalable por el crecimiento de datos respecto a su velocidad o volumen.

NIST

NIST es el acrónimo de National Institute of Standards and Technology, institución americana que estudia, define y promueve estándares tecnológicos.

La base de esta clasificación es poder identificar correctamente estos dos escenarios ya sea mediante recursos internos o con la ayuda de especialistas externos. La combinación de estos dos aspectos nos proporciona tres tipologías de *big data* y un escenario en el que no existe esta necesidad. Aunque es patente el valor que aporta *big data*, no todos los problemas de una organización son necesariamente un problema de datos. Los tipos disponibles se resumen en la tabla 2.

Estructura de datos relacional

Cuando hablamos de estructura de datos relacional hacemos referencia a un tipo de base de datos que permite establecer interconexiones o relaciones entre los datos guardados en tablas.

Tabla 2. Tipos de *big data*

Tipo	Descripción
Tipo 1	Donde una estructura de datos no relacional es necesaria para el análisis de negocio
Tipo 2	Donde es necesario aplicar estrategias de escalabilidad horizontal para procesar y analizar de manera eficiente el negocio
Tipo 3	Donde es necesario procesar una estructura de datos no relacional mediante estrategias de escalabilidad horizontal para procesar y analizar de manera eficiente el negocio

Fuente: NIST.

Por tanto, para una determinada necesidad de negocio, es posible identificar si estamos en un escenario de *big data* o no, y si es necesario este tipo de tecnologías, hecho que cada vez más se erige como un punto relevante y de partida para la implementación de este tipo de proyectos. Esto se resume en la tabla 3.

Tabla 3. Autoevaluación de la existencia de *big data*

Volumen	Velocidad	Variedad	Escalabilidad horizontal	Estructura no relacional	Tipo de *big data*
No	No	No	No	No	No
No	No	Sí	No	Sí	Sí, tipo 1
No	Sí	No	Sí	Quizá	Sí, tipo 2
No	Sí	Sí	Sí	Quizá	Sí, tipo 3
Sí	No	No	Sí	Quizá	Sí, tipo 2
Sí	No	Sí	Sí	Sí	Sí, tipo 3
Sí	Sí	No	Sí	Quizá	Sí, tipo 2
Sí	Sí	Sí	Sí	Sí	Sí, tipo 3

Fuente: NIST.

Escalabilidad

Cuando hablamos de escalabilidad nos referimos a la habilidad de un sistema, red o proceso para reaccionar y adaptarse sin perder calidad, o para manejar el crecimiento continuo de trabajo de manera fluida, o para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. La escalabilidad horizontal está fundamentada en el incremento de nodos del sistema, proceso o red, mientras que la vertical consiste en añadir más recursos (memoria, disco duro y/o procesadores).

Esta tabla permite evaluar una necesidad de negocio. Una reflexión interesante es que dentro de una misma organización pueden plantearse diferentes escenarios de *big data* para resolver distintas necesidades de negocio, lo que en definitiva apunta que serán necesarias diferentes tecnologías de *big data*.

2.2.2. Estándares en *big data*

A medida que *big data* ha adquirido mayor importancia para las organizaciones y estas se han empezado a preocupar por cómo llevar a cabo un proyecto de este tipo, ha quedado patente que se necesita interconectar múltiples sistemas y tecnologías. Esta integración e interoperabilidad de sistemas requiere estándares de mercado.

Por ejemplo, dentro del contexto de la inteligencia de negocio y la analítica ya existen estándares como UIMA (*Unstructured Information Management Architecture*), OWL (*Web Ontology Language*), PMML (*Predictive Model Markup Language*), RIF (*Rule Interchange Format*) y XBRL (*eXtensible Business Reporting Language*), que permiten la interoperabilidad de analítica de datos en información no estructurada, ontologías de modelos de datos, modelos predictivos, el intercambio de datos entre organizaciones y reglas e informes financieros respectivamente.

Desde 2012, varios grupos de trabajo de la comunidad internacional han empezado a trabajar en la creación de estándares, por ejemplo NIST –del que ya hemos hablado con anterioridad–, TMForum, Cloud Security Alliance, ITU, Open Data Platform initiative (ODPi) y Common Criteria Portal.

En el caso de ODPi, su búsqueda de estándares se fundamenta en proponer una configuración mínima de Apache Hadoop que según su criterio incluye solo cuatro componentes: HDFS, YARN, MapReduce y Ambari.

La gran mayoría de estos grupos soporta la adopción efectiva de tecnología *big data* a través del consenso en definiciones, taxonomías, arquitecturas de referencia, casos de uso y *roadmap* tecnológicos.

Taxonomía

Cuando hablamos de taxonomía hacemos referencia a una clasificación u ordenación en grupos de cosas que tienen unas características comunes.

Además, dentro de los proyectos de *big data* es natural encontrar diferentes tecnologías que cumplen diferentes estándares/teoremas, como ACID, CAP o BASE. ACID es el estándar *de facto* de las bases de datos relacionales; CAP, de los sistemas distribuidos; y BASE se está estableciendo para las tecnologías *big data*.

ACID es el acrónimo de *atomicity* (atomicidad), *consistency* (consistencia), *isolation* (aislamiento) y *durability* (durabilidad). Para comprenderlo, debemos definir cada concepto:

- ***Atomicity***: es el principio de todo o nada: o bien la tarea (o todas las tareas) dentro de una transacción se lleva a cabo o bien no se produce ninguna. Si un elemento de una transacción falla, falla toda la transacción.

- ***Consistency***: la transacción debe cumplir con todos los protocolos o reglas definidas por el sistema en todo momento. La transacción no viola los protocolos, y la base de datos debe permanecer en un estado coherente al principio y al final de una transacción; nunca hay transacciones a medio terminar.

- ***Isolation***: ninguna transacción tiene acceso a cualquier otra transacción que está en un estado intermedio o sin terminar. Por lo tanto, cada operación es independiente. Esto es necesario para el rendimiento y la consistencia de las transacciones dentro de una base de datos.

- ***Durability***: una vez que se complete la transacción, persistirá completa y no se puede deshacer; sobrevivirá a fallos del sistema, pérdida de energía y otros tipos de averías del sistema.

CAP es el acrónimo de *consistency* (consistencia), *availability* (disponibilidad) y *partition tolerance* (tolerancia a la partición). Definimos estos conceptos:

- ***Consistency***: se refiere a si un sistema funciona de manera completa o no. Es decir, ante un error que afecta al dato, se revierte a un estado anterior en el que el dato cumple con las condiciones del sistema. Es la misma condición que en ACID.

- ***Availability***: hace referencia a la disponibilidad del servicio o sistema cuando es solicitado.

- ***Partition tolerance***: representa el hecho de que un sistema dado sigue funcionando incluso bajo circunstancias de la pérdida de datos o fallo del sistema. Un fallo de un nodo dado no debería causar que todo el sistema se colapse.

BASE es el acrónimo de *basically available* (básicamente disponible), *soft state* (estado blando) y *eventual consistency* (consistencia eventual). Definimos estos conceptos:

- ***Basically available***: esta restricción establece que el sistema garantiza la disponibilidad de los datos en lo que respecta a CAP; habrá una respuesta a cualquier solicitud. Sin embargo, esa respuesta aún podría ser «fracaso» para obtener los datos solicitados o los datos pueden estar en un estado incoherente o cambiar.

- ***Soft state***: el estado del sistema podría cambiar con el tiempo, incluso sin entrada de datos, motivados por la consistencia eventual.

- ***Eventual consistency***: el sistema se convertirá eventualmente en constante una vez que se deja de recibir entrada de datos.

Como veremos en el apartado 3, existen muchas tecnologías dentro del contexto de *big data*. Esta proliferación de tecnologías aumenta su complejidad. Por ello, han emergido varias comparativas (*benchmarking*) que permiten evaluar estas tecnologías considerando condiciones iguales. Existen muchas, pero destacamos las comparativas TPC-DS y TPCx-HS por su imparcialidad, así como Big Data Benchmark, de la Universidad de Berkeley, por los casos contemplados.

Lectura complementaria

R. Han; **X. Lu** (2014). «On Big Data Benchmarking». *Big Data Benchmarks, Performance Optimization, and Emerging Hardware. Springer International Publishing* (págs. 3-18).

2.3. ¿Cuándo es necesario *big data*?

TPC

TPC es un organismo que tiene el objetivo de diseminar comparativas verificables sobre el rendimiento de bases de datos. Más información en: http://www.tpc.org.

La no existencia de una definición formal, una que permita distinguir de manera completamente precisa cuándo una organización está en una situación de necesidad de *big data*, ha generado barreras de adopción a este tipo de tecnologías.

Hemos visto que hay escenarios en los que no es suficiente trabajar con un *data warehouse*. También tenemos una clasificación para los tipos de *big data* que permite dirimir escenarios genéricos. Sin embargo, esto no es suficiente en el contexto de una organización donde la experimentación no tiene un amplio margen.

En esta aproximación más pragmática a *big data*, las organizaciones se están trabajando en cinco grandes categorías de casos de uso. Estos casos de uso son movimientos organizacionales de una estrategia de negocio enfocada a *big data*, y son los siguientes:

1) Toma de decisiones

2) Operaciones e inteligencia operacional

3) Validación de hipótesis y resolución de problemas

4) Productos y servicios basados en datos

5) Comercio de datos

Vamos a explicar en detalle cada uno de estos movimientos organizacionales.

2.3.1. Toma de decisiones

El primer caso de uso es la toma de decisiones. Esta aproximación consiste en la ampliación de las capacidades tradicionales de toma de decisiones mediante las tecnologías de *big data*, lo que significa que los sistemas de inteligencia de negocio y almacenes de datos corporativos pueden alimentarse o combinarse con los repositorios de *big data*.

Caso: NH

NH es una cadena hotelera con más cuatrocientos hoteles en veinticinco países. Dentro de la estrategia de mejorar el servicio para sus clientes, la compañía cada año selecciona varios hoteles sobre los que hará mejoras. Las mejoras que se realizan van desde la ampliación del personal hasta la creación de nuevas instalaciones. Para tomar la decisión de «dónde es necesario invertir en este periodo», NH se ha fundamentado tradicionalmente en dos fuentes de datos:

- Los datos financieros consolidados en el *data warehouse* de la compañía.

- Una serie de encuestas realizadas a los clientes para conocer su satisfacción de los servicios e instalaciones del hotel. Estas encuestas no son exhaustivas y no cubren todos los hoteles ni todos los clientes por los costes asociados con su realización.

En los últimos años, en NH se han dado cuenta de que para conocer la satisfacción del cliente, así como las áreas de mejora, la información relevante se encuentra más allá del perímetro de la organización. Los clientes de NH comparten sus impresiones a través de diferentes canales, como pueden ser TripAdvisor, Yelp o Expedia. Es decir, estamos hablando de fuentes de datos externas a la organización y además no estructuradas o con diferentes formatos.

El enfoque de la organización ha sido complementar la información financiera en el *data warehouse* con información externa que se almacena y se procesa con tecnologías de *big data* y minería de texto (para extraer los comentarios relevantes para la mejora de los hoteles), y que permite mejorar y complementar la toma de decisiones en un proceso ya existente.

2.3.2. Operaciones e inteligencia operacional

El segundo caso de uso son las operaciones y la inteligencia operacional, que suceden en tiempo real. Esta aproximación consiste en la aplicación de estas tecnologías en el ámbito de operaciones tanto para el control y el análisis de proceso de negocio como para el diseño e implementación de sistemas transaccionales. Este segundo escenario trasciende de la toma de decisiones y permite entender por qué las tecnologías *big data* están llamadas a ser muy relevantes dentro de las tecnologías de información. Es previsible que se integren de modo natural en múltiples aplicaciones.

Por lo tanto, estamos hablando, por un lado, de sistemas de inteligencia y detección de patrones en tiempo real y, por el otro, de sistemas operacionales que o bien por sus necesidades en escalabilidad o bien por su complejidad en el esquema de los datos ya no se fundamentan en tecnologías relacionales.

Caso: Santander/Caixabank

Uno de los puntos más relevantes para muchas organizaciones es cuando interaccionan con sus clientes. Es lo que llamamos momentos de la verdad. Entre estos momentos destaca cuando el cliente se pone en contacto con una organización para la resolución de una incidencia. En épocas anteriores se ha automatizado el proceso (mediante sistemas de respuesta predefinida) o se ha dividido el servicio en distintas capas dentro y fuera de la organización para tener diferentes niveles de servicio.

En el contexto financiero son varias las entidades financieras españolas que ya usan *speech analytics* para entender las emociones de sus clientes durante sus interacciones en una llamada. Es posible, por lo tanto, detectar cuándo va a disminuir la satisfacción del cliente y actuar consecuentemente.

2.3.3. Validación de hipótesis y resolución de problemas

Uno de los escenarios más importantes es la validación de hipótesis y resolución de problemas. Este escenario consiste en encontrar soluciones para problemas de negocio que no han sido anteriormente abordados en una organización y para los cuales no hay preguntas predefinidas. Es decir, se busca conocer qué ha sucedido, qué factores son los más relevantes y el porqué. Es necesario crear hipótesis y validarlas a través de la técnica más adecuada y eficiente. Este

tipo de aplicación es el equivalente a tener a Sherlock Holmes en casa. Es, en definitiva, un entorno que debe ser lo suficientemente flexible para funcionar en diferentes escenarios de necesidades.

El resultado puede ser una solución puntual o una propuesta que pase a convertirse en uno de los otros escenarios.

Caso: Sky

Sky es una empresa que produce y distribuye contenidos de vídeo tanto en directo como bajo demanda con presencia en distintos países europeos. La distribución de contenidos de vídeo se realiza a través de redes informáticas conocidas como *content delivery networks*. Estas redes están formadas por múltiples elementos tanto pertenecientes a la propia compañía como a terceros que deben asegurar que la distribución se realiza manteniendo el nivel de calidad contratado por el cliente. Frecuentemente, el proceso de transmisión de vídeo a través de la red se controla y se mide para asegurar su correcto funcionamiento. Este es el caso de Sky. Sin embargo, meses atrás tuvo un gran fallo en su *content delivery network* durante la transmisión de la jornada futbolística en el fin de semana, que producía errores de acceso al sistema, congelación de la imagen o transmisión de imágenes en baja calidad. A pesar de tener un sistema de monitorización, Sky no conocía los motivos por los que había sucedido esta caída de calidad en el servicio.

El enfoque de la organización ha sido contratar a un experto para investigar lo sucedido, lo que se traduce en este caso en trabajar con millones de registros en formato *log* e investigar las causas del error. Tras aplicar lo que se conoce como *root cause analysis*, una técnica para encontrar de forma sistemática los factores que provocan un fallo, a un conjunto de datos almacenados en tecnologías de *big data* por su tamaño, se encontraron las razones del error y se propusieron una serie de mejoras para la red.

2.3.4. Productos y servicios de datos

El cuarto escenario de uso es la creación de productos y servicios basados en datos. El dato se transforma en la pieza angular para mejorar la experiencia de uso del producto y servicio o para el diseño y despliegue de este.

Estamos hablando de modelos de negocio en los que el dato y los algoritmos analíticos generan valor tanto para el cliente como para la organización, y por ello modifican todos los aspectos primordiales del modelo de negocio.

Caso: Nest

Nest es una empresa que produce dispositivos inteligentes, en particular termostatos, detectores de humo y cámaras de vigilancia, entre otros. Fue adquirida por Google en 2014.

Los productos creados por Nest son un ejemplo de producto basado en datos y algoritmos. Por ejemplo, el termostato contiene diferentes tipos de sensores para detectar a las personas y los animales presentes en el hogar, así como la temperatura de este. Además, va registrando las preferencias de las personas que viven en casa. A saber, a qué hora están en casa, cuántas personas, en qué habitaciones se encuentran y cuál es la temperatura a la que prefieren su casa. Y lo combinan con información contextual como dónde se encuentra la casa y en qué época del año. Con todos estos datos, se crea un perfil de preferencias y, a partir de un cierto momento, el dispositivo empieza a trabajar de manera automática. Este proceso automático permite reducir el coste energético.

Existe también otro potencial beneficiario de estos datos, aunque en formato agregado: las empresas productoras y distribuidoras de energía. A partir de los datos de todos los usuarios de Nest en aquellas zonas geográficas en las que ofrecen servicio, pueden conocer las necesidades energéticas y, por lo tanto, ajustar la demanda.

2.3.5. Comercio de datos

El último escenario de uso es el comercio de dato. El dato se prepara para su venta a terceros. Esto puede incluir diversos procesos, como agregación, transformación y distribución del dato o, en el caso de contener información sensible, enmascarar dichos datos para que el conjunto final contenga datos anónimos. Este tipo de uso también puede derivar en diseñar una plataforma *ad hoc*. El dato puede comercializarse en bruto o en forma de conocimiento.

Caso: Vodafone/TomTom

Vodafone es una conocida compañía que proporciona servicios de telecomunicaciones a nivel mundial. TomTom es una compañía que ofrece productos y servicios de GPS.

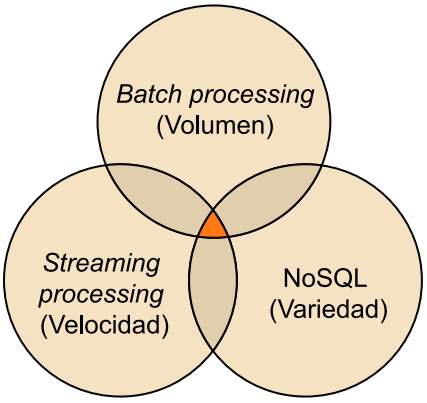
Los servicios de TomTom permiten conocer la ruta óptima a un conductor. La calidad de este servicio depende de trabajar con datos actualizados y contextualizados, incluyendo accidentes o atascos de tráfico. Por ello, TomTom compra datos de terceros, como Vodafone.

En este caso particular, Vodafone comercializa los datos agregados, anónimos y geolocalizados de los usuarios de su red. En el caso de tener una gran acumulación de usuarios en un mismo lugar (y estar dicho lugar en una carretera), esto se traduce en una situación de atasco o accidente. Por ello, TomTom puede usar esta información para proponer una ruta alternativa y ofrecer una mejor experiencia de cliente.

3. Tecnologías de *big data*

Un primer enfoque para pensar en las tecnologías de *big data* es recuperar las 3 V presentadas en el subapartado 1.2. Diferentes problemáticas del dato necesitan diferentes paradigmas, tal y como apuntan Casado y Younas, y como se ilustra en la figura 2.

Figura 2. Tecnologías de *big data*



Fuente: R. Casado y M. Younas.

Lectura Complementaria

R. Casado; M. Younas (2015). «Emerging trends and technologies in big data processing». *Concurrency and Computation: Practice and Experience* (núm. 27(8), págs. 2078-2091).

Existe una correspondencia directa entre la explosión en la problemática en el dato y la emergencia de una determinada tecnología. De esta manera, tenemos:

1) Tecnologías de procesamiento por lotes o *batch processing*: permiten resolver problemas vinculados con el volumen del dato.

> Criteo es una compañía que ofrece soluciones para marketing digital basadas en datos. Esta organización usa las tecnologías de procesamiento por lotes para consolidar datos y optimizar sus algoritmos de análisis de campañas de marketing.

2) Tecnologías de procesamiento en flujo o (*streaming processing*): permiten resolver problemas vinculados con la velocidad del dato.

> Capital One es una entidad bancaria que ofrece productos y servicios financieros a consumidores. Esta organización usa las tecnologías de procesamiento en flujo para monitorizar la actividad de sus clientes en tiempo real.

3) NoSQL: permiten resolver problemas vinculados con la variedad del dato.

Metlife es una entidad aseguradora con presencia internacional. Usa las tecnologías NoSQL para integrar todas las referencias de cliente en un único punto de acceso y tener una visión de 360 grados.

NoSQL

Cuando hablamos de NoSQL hacemos referencia a bases de datos no relacionales. NoSQL es el acrónimo de *not only SQL*. SQL es el acrónimo de *Structure Query Language* y hace referencia al lenguaje de consultas de bases de datos relacionales.

Esta aproximación a las tecnologías de *big data* está principalmente centrada en dos puntos: el almacenamiento y el procesamiento. En este material vamos a ampliar los puntos a tratar añadiendo el análisis y la visualización. El motivo detrás de este enfoque reside en que los cambios en las capas de procesamiento y almacenamiento influyen en el resto.

Cuando hablamos de tecnologías de *big data* en realidad nos estamos refiriendo a una colección de componentes, plataformas y soluciones que cubren las diferentes necesidades para con el dato. Estas necesidades son las siguientes:

- **Almacenamiento**: permitir el almacenamiento del dato conforme a las necesidades de negocio.
- **Procesamiento**: permitir la captura, la transformación y el movimiento del dato conforme a las necesidades de negocio.
- **Análisis**: permitir la generación de valor para el negocio a partir del dato.
- **Visualización**: permitir la presentación y comunicación de los resultados de acuerdo con las necesidades de negocio.

Frecuentemente estas tecnologías se combinan en ecosistemas, como discutiremos en el subapartado 4.3, y formando parte de una arquitectura, como se explicará en el subapartado 4.1.

Muchas de las tecnologías de *big data* tienen origen *open source* para acelerar la innovación, lo que significa que podemos tener acceso a una versión *community* y, al mismo tiempo, varios fabricantes ofrecen una plataforma de pago con diferentes componentes integrados y preparados a nivel empresarial.

3.1. Almacenamiento

HPC

Cuando hablamos de HPC nos referimos a la práctica de añadir capacidad de computación de manera que mejora el rendimiento de una estación de trabajo y es posible abordar problemas complejos en la ciencia, ingeniería y/o negocios.

En las últimas décadas, las bases de datos relacionales han sido la opción de almacenamiento *de facto* para los sistemas de información. En algunos contextos con grandes necesidades de almacenamiento y procesamiento, como puede ser la meteorología, se ha trabajado con sistemas combinados de hardware y software optimizados para tareas intensivas en el dato, conocidos como *high performance computing* (HPC). El enfoque de HPC se ha fundamentado principalmente en la escalabilidad vertical.

Con la emergencia de *big data* esto está cambiando de manera significativa, principalmente por varios motivos:

- La tecnología relacional no es escalable para soportar el volumen de datos en el contexto de *big data*.

- La tecnología relacional es incompatible con los datos no estructurados, que cada vez son más relevantes para el negocio.

- La nueva tecnología no necesita de HPC para ejecutarse, sino que puede trabajar con redes de ordenadores trabajando de manera combinada con prestaciones de computación menores individualmente pero mayores colectivamente.

SLA

Cuando hablamos de SLA nos referimos a un acuerdo que estipula el nivel de servicio, el soporte, las posibles penalizaciones, el nivel de alta disponibilidad tanto de hardware como de software y el precio.

En el contexto de un proyecto de *big data* existen diferentes tecnologías de almacenamiento que habilitan estrategias eficientes y escalables tanto en coste como en respuesta a las necesidades de la naturaleza del dato. Una de las características de este tipo de sistemas es que proporcionen alta disponibilidad (*high availability* o HA) y/o tolerancia a fallos (*fault tolerance* o FT). Aunque similares, no son lo mismo. Por un lado, HA implica tener un esquema en el que los tiempos de caídas deben mantenerse muy cortos en un periodo anual. Esto se traduce en mantener un sistema de SLA (*service level agreement*) muy elevado. Por otro lado, FT hace referencia a un sistema donde no existe la posibilidad de perder ni un solo minuto de trabajo en producción, lo que implica tener infraestructura totalmente redundante.

Una de las técnicas usadas para la alta disponibilidad es la replicación que habilita la copia y el mantenimiento de los objetos en una base de datos distribuida. También se conoce como *sharding*. Usaremos indistintamente una palabra u otra.

La tabla 4 resume las diferentes opciones disponibles y qué aporta cada una de ellas.

Tabla 4. Tecnologías del almacenamiento

Tecnología	Descripción	Características	Productos	Caso de uso
Sistema de archivos distribuido	Sistema que proporciona almacenamiento basado en la división de los datos en ficheros y servidores	Proporciona redundancia y alta disponibilidad por replicación. Acceso secuencial de datos. Para minimizar las lecturas de búsqueda a disco, así como el procesamiento de muchos ficheros, este tipo de sistemas agrega los datos en ficheros de mayor tamaño.	Apache HDFS, Amazon S3 o Google File System	Archivado de conjuntos de datos. Almacenamiento de datos en bruto. Almacenamiento de bajo coste para largos periodos.
NoSQL	Sistema que proporciona almacenamiento basado una ordenación/representación no relacional	En general cumple: escalado horizontal en lugar de vertical; alta disponibilidad, consistencia eventual; BASE, no ACID y *auto-sharding*. Persistencia políglota. Consultas distribuidas.	MongoDB, Apache Cassandra, Riak, Redis, Neo4j o CouchDB	El modelo de negocio no puede representarse de forma relacional. El modelo de negocio evoluciona rápidamente y necesita una base de datos flexible en su modelo.

Tecnología	Descripción	Características	Productos	Caso de uso
NewSQL	Sistema NoSQL que combina propiedades ACID	Además de las características de NoSQL, incluye soporte para SQL y el uso de estructuras relacionales.	VoltDB, NuoDB, Google Spanner o CockroachDB	Sistemas OLTP con alto volumen de transacciones. Analítica en tiempo real.
In-memory	Uso de la memoria del procesador para el almacenamiento de datos	Reduce la latencia de acceso y de cálculo. Puede basarse en *grid* o base de datos.	HazelCast. Pivotal Gemfire, Aerospike, MemSQL o Altibase HDB	Analítica operacional. BI operacional. *Streaming analytics*.

El sistema de archivos distribuido también ha sido adoptado por las bases de datos relacionales, lo que da lugar a poder trabajar en paralelo y que se conoce como *massive parallel processing* (MPP). Tenemos ejemplos como: Teradata, IBM Netezza, Pivotal Greenplum u Oracle Exadata.

Dentro de NoSQL, existen principalmente cuatro tipos de bases de datos:

1) *Key-value store*: el almacenamiento se fundamenta en el uso de parejas clave-objeto en las que no hay esquema alguno. Ejemplos: Apache HDFS, Riak, Voldemort, Redis, RocksDB o Amazon DynamoDB.

2) Bases de datos orientadas a columnas: el almacenamiento del dato se realiza por columnas, no por filas. Ejemplos: Apache Hbase, Apache Cassandra, MonetDB, Druid, HP Vertica, SAP IQ, LucidDB, ScyllaDB o Amazon SimpleDB.

3) Bases de datos de grafos: usa nodos y vértices para representar datos. Ejemplos: Neo4J, HyperGraphDB, ArangoDB, Ontotext GraphDB u OrientDB.

4) Bases de datos orientadas a documentos: el almacenamiento del dato se realiza como si fuera un documento semiestructurado. Ejemplos: MongoDB, CouchDB o MarkLogic.

Grafo

Cuando hablamos de grafo nos referimos a un conjunto de objetos (llamados vértices o nodos) unidos por enlaces (llamados aristas o arcos). Un grafo permite estudiar las interrelaciones entre sus nodos.

Para algunas de las opciones disponibles, las distinciones entre las diferentes bases de datos se están diluyendo, ya sea porque una misma base de dato pasa a ser multi-NoSQL (soportando más de un tipo) o porque pertenece a varias de las categorías al mismo tiempo. Ejemplos: ArangoDB combina grafos, documentos y *key-value*; OrientDB combina grafos, documentos, objetos y *key-value*. En general, estamos hablando de una alta especialización en el caso de uso y, por lo tanto, de un escenario políglota en el almacenamiento del dato.

Los avances en tecnologías *in-memory* y NewSQL han abierto la puerta a sistemas que permiten combinar en una única base de datos OLTP (*online transaction processing*) y OLAP, lo que se conoce como *hybrid transactional/analytical processing* (HTAP), término acuñado por Gartner. Productos como SAP HANA, MemSQL, VoltDB, NuoDB y OrientDB soportan este enfoque.

3.2. Procesamiento

La necesidad de procesar datos no es un aspecto nuevo para las organizaciones. En el pasado se ha abordado usando técnicas de integración de datos, o *data integration*. Aunque existen muchas técnicas de integración, el procesamiento de *big data* se fundamenta principalmente en ELT (*extract*, *load*, *transform*). Es decir, se pone foco en guardar el dato en bruto, con el menor número de cambios, y el proceso de transformación se ejecuta en cada una de las bases de datos (sea cual sea su tipología).

Data integration

Cuando hablamos de *data integration* hacemos referencia al conjunto de aplicaciones, productos, técnicas y tecnologías que permiten una visión única y consistente de nuestros datos de negocio.

En línea con los sistemas de almacenamiento, las principales aproximaciones para el procesamiento son las siguientes:

- **Procesamiento de datos en paralelo**: lo que significa que un proceso se divide en múltiples tareas que se ejecutan en paralelo. Tradicionalmente este enfoque se ha realizado con una única máquina con múltiples procesadores o núcleos.

- **Procesamiento de datos distribuidos**: lo que significa que el proceso se divide en múltiples tareas que se ejecutan en un clúster de máquinas conectadas en red siguiendo la filosofía «divide y vencerás».

Clúster

Cuando hablamos de clúster nos referimos al conjunto de ordenadores conectados en red que trabajan de manera conjunta. Cada ordenador del clúster es llamado nodo. Si los ordenadores son heterogéneos, realizan tareas independientes o no están en la misma localización, hablamos de *grid*.

En el contexto de *big data*, para poder abordar las necesidades de trabajar con grandes volúmenes de datos y/o de capturarlos y consumirlos a diferentes velocidades (desde horas hasta por debajo del segundo), han emergido diferentes aproximaciones:

1) Procesamiento en modo *batch*, o por lotes: el dato se procesa en modo *offline*. Su latencia puede ir desde minutos hasta horas. El dato se ha almacenado previamente antes de ser procesado. Apache MapReduce y Spark, este último con mejores prestaciones en término de velocidad, permiten este tipo de procesamiento.

> Hulu, un servicio de vídeo en *streaming* con más de cinco millones y medio de suscriptores y más de veinte millones de visitantes únicos por mes, usa MapReduce para procesar los *logs* resultado de la visualización de más de cuatrocientos millones de vídeos al mes. El objetivo es poder ofrecer un servicio de *streaming* con un nivel de calidad consistente. Es decir, siempre disponible, desde cualquier dispositivo y con el nivel de calidad de vídeo adecuado al dispositivo.

2) Procesamiento en modo *real time*, o en tiempo real: el dato se procesa en modo *online*. Su latencia está en el rango desde menos de un segundo hasta el minuto. Por ello, el dato se procesa en memoria en el momento de su captura antes de almacenarlo. Hay dos tipos: procesamiento en flujo (*stream*), en el que el dato llega de forma continua, y procesamiento por intervalos o eventos (*event*). Apache Storm, Apache Flink y Spark permiten este tipo de procesamiento.

MyFitnessPal es un servicio que permite conocer el número de calorías consumidas y da soporte al tratamiento de dietas. Esta compañía usa Spark para limpiar, mejorar y complementar los datos específicos de comida introducidos por los usuarios con el objetivo de tener una base de datos de comida/calorías de máxima calidad en tiempo real. Es importante para este servicio que sea lo más cómodo y menos intrusivo para el usuario. Además, también se aprovecha de las capacidades de Spark para hacer recomendaciones.

El procesamiento por intervalos no es nuevo. Los sistemas CEP (*complex event processing*) se han usado desde hace años en sectores como la banca o la energía para resolver esta necesidad. En este tipo de sistemas, lo relevante no es procesar todo el flujo de datos, sino detectar aquellos subconjuntos que cumplen un patrón. El sistema monitoriza el flujo de datos y lo compara con los patrones definidos, por ejemplo, para detectar el fraude en entidades financieras. Lo relevante es detectar que un determinado cliente está realizando un conjunto de operaciones sospechosas de cometer un fraude.

CEP

Cuando hablamos de CEP hacemos referencia al procesamiento de eventos en tiempo real que combinan múltiples fuentes y que se usa para inferir eventos o patrones que siguieren situaciones complicadas como oportunidades y/o amenazas.

El teorema CAP está vinculado con el procesamiento en lotes, mientras que el teorema SCV es el que aplica al procesamiento en tiempo real. La relación de los tres componentes de SCV se ilustra en la figura 3 y significa:

- Si se necesita S y C, no es posible procesar grandes volúmenes de datos porque retrasan el procesamiento.
- Si se necesita C y V, no es posible trabajar a una gran velocidad porque el procesamiento a gran velocidad requiere menores cantidades de datos.
- Si se necesita V y C, se consideran muestras (en lugar de trabajar con todo el conjunto de datos), lo que reducirá la consistencia.

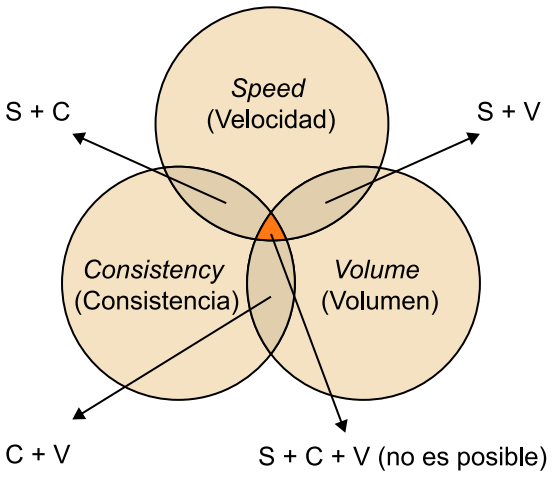
SVC

Cuando hablamos de SCV hacemos referencia a *speed, consistency* y *volume*. Es decir, a la velocidad de procesamiento, a la exactitud del dato y a la cantidad de datos procesados.

Machine learning (ML)

Cuando hablamos de *machine learning* (ML) hacemos referencia a una rama de la informática que ha evolucionado desde el estudio y reconocimiento de patrones hacia la inteligencia artificial. Se fundamenta en diferentes tipos de algoritmos clasificados en aprendizaje supervisado, no supervisado y basados en refuerzos.

Figura 3. SCV



Fuente: Josep Curto.

Cabe comentar que también han emergido motores de procesamiento especializados y optimizadas, como H_20 o la versión distribuida de TensorFlow con foco en *machine learning*. Por ejemplo, en el caso de H_20 incluye capacidades de procesamiento distribuido en memoria para los algoritmos analíticos soportados, pero no para tareas genéricas como puede ser Hadoop o Spark.

3.3. Análisis

Como se ha comentado anteriormente, lo más importante para una organización no es ser capaz de almacenar o procesar datos, sino generar valor a partir de ellos. El valor toma la forma del análisis. La creciente complejidad en el dato ha permeado en la capa del análisis, lo que significa ajustar y modificar los diferentes tipos de análisis a la nueva naturaleza del dato.

El análisis se concentra al final en dos grandes áreas: inteligencia de negocio, dirigida a conocer el rendimiento pasado, y analítica de negocio, enfocada a predecir el rendimiento futuro y conocer patrones ocultos en el dato.

Para distinguir claramente los cambios en procesamiento y almacenamiento, hemos separado el *data warehouse* y la integración de datos de la inteligencia de negocio aunque, en general, no se conciben este tipo de sistemas sin estos componentes. Sin embargo, *big data* abre la puerta a una nueva combinación y de ahí la separación que estamos considerando, puesto que la arquitectura para el almacenamiento y el procesamiento de datos puede llegar a ser más compleja de lo que era antaño, como se discutirá en el subapartado 4.1. Es necesario recordar los diferentes componentes de la inteligencia de negocio:

- **Informes**: documentos a través de los cuales se presentan los resultados de uno o varios procesos de negocio que pueden distribuirse o simplemente estar disponibles para su acceso. Suelen contener texto acompañado de elementos como tablas o gráficos para agilizar la compresión de la información presentada.

- **OLAP (*online analytical processing*)**: método para organizar y consultar datos sobre una estructura multidimensional.

- **Cuadros de mando (o *dashboard*)**: sistema que informa de la evolución de los parámetros fundamentales de negocio de una organización o de un área de esta a través de componentes visuales integrados.

- ***Scorecards***: tipo de cuadro de mando formado solo por listas de indicadores. A veces también toma la forma de informe.

- **Consultas *ad hoc***: método que ofrece autoservicio y exploración de datos a usuarios finales basados en metadatos de negocio.

Lectura complementaria

R. S. Kaplan; D. P. Norton (1996). *The Balanced Scorecard: Translating Strategy into Action*. Boston, MA: Harvard Business School Press.

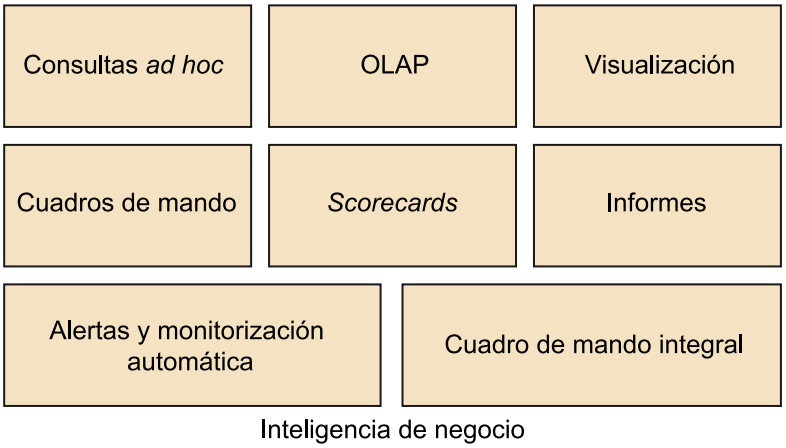
- **Alertas y monitorización automática**: sistema para crear, gestionar y distribuir alertas críticas basadas en indicadores clave de negocio con foco en la gestión de excepciones.

- **Cuadro de mando integral (o *balanced scorecard*)**: método de planificación estratégica basado en métricas y procesos ideado por los profesores Kaplan y Norton, que relaciona factores medibles de procesos con la consecución de objetivos estratégicos.

Una solución de inteligencia de negocio puede tener solo uno o varios de estos componentes. Las soluciones más maduras de mercado suelen tener todos en formato modular. La implementación de uno o más componentes en una organización debe depender de las necesidades de negocio en la organización, y no de la plataforma, del proveedor seleccionado o de las preferencias del usuario de negocio o departamento.

La figura 4 ilustra los componentes de la inteligencia de negocio.

Figura 4. Inteligencia de negocio



Fuente: Josep Curto.

Consideremos un ejemplo.

> Jagex es una compañía de videojuegos para móviles que soporta millones de usuarios jugando al mismo tiempo. Para esta compañía es absolutamente primordial comprender a sus clientes: aquellos que pagan, aquellos que se dan de alta, qué productos virtuales se compran y se usan en cada uno de los videojuegos, y poder analizar esta información tanto temporal como geográficamente. Para ello, se han combinado las capacidades de almacenamiento y procesamiento de *big data* con las capacidades de análisis en formato cuadro de mando e informes de la inteligencia de negocio para tener control del negocio.

En la analítica de negocio también tenemos diferentes tipos de análisis. Destacamos los siguientes, aunque no es una taxonomía exhaustiva ni exenta de solapamientos:

- **Análisis estadístico/cuantitativo**: rama de las matemáticas que investiga la recopilación, el análisis, la interpretación y la presentación de datos de una muestra representativa, busca explicar las correlaciones y dependencias de un fenómeno físico o natural, de ocurrencia en forma aleatoria o condicional. El análisis cuantitativo es un conjunto de técnicas de análisis estadístico que puede incluir, entre otros, el análisis cuantitativo del comportamiento.

- **Minería de datos**: técnica que permite la extracción de información y conocimiento a partir del dato.

- **Minería de textos**: técnica que permite la extracción de información y conocimiento a partir de texto.

- **Minería de procesos**: técnica que permite el análisis de los procesos de negocio basado en *logs* de eventos.

- ***Machine learning***: conocida también como aprendizaje automático, es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

- **Inteligencia artificial**: área multidisciplinar que combina computación, matemáticas, lógica, etc., y que busca el diseño de sistemas capaces de resolver problemas cotidianos por sí mismos, utilizando como paradigma la inteligencia humana.

- **Analítica de contenidos**: técnica que permite la extracción de información y conocimiento de contenido, como puede ser imágenes o vídeos. El foco no solo está en la extracción de valor, sino también en la composición automática de contenidos personalizados.

- **Analítica de grafos**: técnica que permite la extracción de información y conocimiento de datos estructurados como un grafo.

- **Analítica visual**: técnica que habilita la exploración de datos y la detección de patrones a través de técnicas de visualización.

- **Modelización predictiva**: técnica para la representación de modelos mediante técnicas estadísticas o matemáticas (como ecuaciones diferenciales) que permite identificar representaciones y hacer predicciones.

La estadística, la inteligencia artificial, el *machine learning* y la minería de datos y texto son disciplinas relacionadas y, en realidad, habilitan los casos de uso presentados en esta taxonomía.

La figura 5 ilustra los componentes en la analítica de negocio.

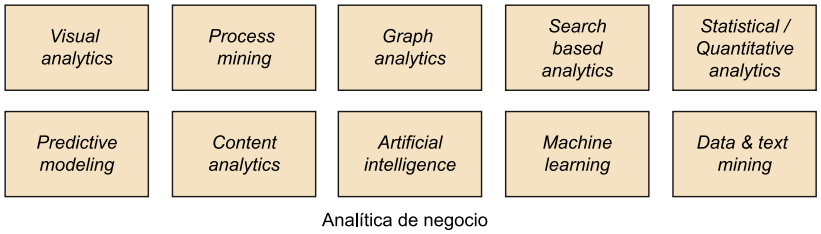
Inteligencia artificial

Dentro de los sistemas que buscan simular la inteligencia humana, destacan los sistemas cognitivos. La computación cognitiva hace referencia a hardware y software que simula el funcionamiento del cerebro humano para tomar decisiones. El aprendizaje se fundamenta en instrucciones y experiencia.

Ecuaciones diferenciales

Cuando hablamos de ecuaciones diferenciales estamos haciendo referencia a una ecuación matemática que relaciona una función y sus derivadas.

Figura 5. Analítica de negocio



Fuente: Josep Curto.

Consideremos un ejemplo.

> SuperCell es una compañía de videojuegos para móviles que soporta millones de usuarios jugando al mismo tiempo. Entre sus éxitos destaca *Clash of Clans*. Esta compañía usa la combinación de tecnologías de almacenamiento de *big data* y analítica de negocio para validar hipótesis de negocio. Uno de los test A/B realizados ha sido para decidir si valía la pena añadir la conectividad de Facebook conociendo si los usuarios usan esta posibilidad para invitar a sus amigos y para compartir sus logros y si esto incide en la retención del usuario.

Las taxonomías presentadas tienen una razón de ser. La principal diferencia de la inteligencia y la analítica de negocio tradicionales con respecto a *big data* es que cada uno de los componentes se ha tenido que adaptar. Por un lado, en el contexto de la inteligencia de negocio esto sucede:

- A través de conectores para el uso de los sistemas de almacenamiento y procesamiento de *big data*.

- A través la adaptación de la tecnología a la complejidad del dato. Tenemos, por ejemplo, Apache Kylin, creado por eBay, que proporciona OLAP para *big data*, Hue, creado por Cloudera, que permite visualizar consultas *ad hoc* sobre Hadoop, o Caravel, de Airbnb, con foco en la exploración de datos.

Aunque nos estamos centrando en nuevas tecnologías y fabricantes, los fabricantes tradicionales como IBM, Microsoft, Microstrategy, Oracle o Information Builders también se están posicionando en este mercado, creando su propia propuesta integrada y/o a través conectores específicos para su plataforma.

Por otro lado, en el contexto de la analítica tenemos también que las soluciones y librerías ya existentes se están adaptando de una manera similar a la inteligencia de negocio. Adaptar en este caso se traduce en crear nuevas versiones del algoritmo que encapsula una cierta técnica para que pueda aplicarse a un conjunto de datos complejos, pueda escalar y, sobre todo, tenga sentido desde un punto de vista estadístico y matemático.

Lectura complementaria

J. Leskovec; A. Rajaraman; J. Ullman (2016). *Mining of Massive Datasets* (2.ª ed.).

Por ello, el gran cambio reside en la aparición de nuevas librerías de *machine learning*, *graph analytics* y *deep learning* adaptadas a *big data*. En el apartado 4 revisaremos los principales componentes de algunos de los principales ecosistemas en *big data* y veremos cómo la gran mayoría de ellos incluyen este tipo de librerías.

Vamos a entrar un poco más en detalle en las librerías vinculadas a *machine learning*. El número se ha multiplicado considerablemente en los últimos años y muchas de ellas son *open source* con el objetivo de crear comunidades de desarrolladores para acelerar su evolución, aunque su origen está vinculado a una determinada plataforma o fabricante. Para permitir el desarrollo de aplicaciones de negocio, normalmente soportan más de un lenguaje de desarrollo (frecuentemente los mismos que soporta la librería de procesamiento a la que están vinculados).

La tabla 5 recoge algunas de las principales librerías.

Tabla 5. Principales librerías de *machine learning*

Nombre	Ecosistema	Lenguajes
Apache Mahout	Apache Hadoop	Java
Spark	Apache Spark	Java, Scala, R y Python
Apache FlinkML	Apache Flink	Java, Scala y Python
H_20	Propio ecosistema (soporte Hadoop y Spark)	Java, Python, R, Scala
TensorFlow	Google	C++, Python
SAMOA	Yahoo (Hadoop)	Java
Oryx	Cloudera (Hadoop)	Java
Weka	Pentaho (Hadoop)	Interfaz/Java
R	Multisistema	Lenguaje R

Esta tabla no entra en el detalle de los diferentes algoritmos soportados dentro de cada categoría, puesto que evolucionan rápidamente. Las librerías seleccionadas, es decir, Apache Mahout, Spark MLlib, Apache FlinkML, H_20, TensorFlow, SAMOA, Oryx, Weka o R son tan solo algunos de los proyectos en esta área. Vinculados con el lenguaje de programación Python tenemos también muchas iniciativas interesantes, como Scikit-learn, Pandas o Caffe, que también tienen adaptadores para tecnologías de *big data* o que directamente son distribuidas. También es necesario seguir de cerca iniciativas que son actualmente *open source*, como CNTK, creada por Microsoft, Aerosolve, de Airbnb, y KeystoneML, creada en UC Berkely AMPLab.

Java, Scala, C++, R y Python

Cuando hablamos de Java, Scala, C++, R y Python estamos haciendo referencia a diferentes lenguajes de programación.

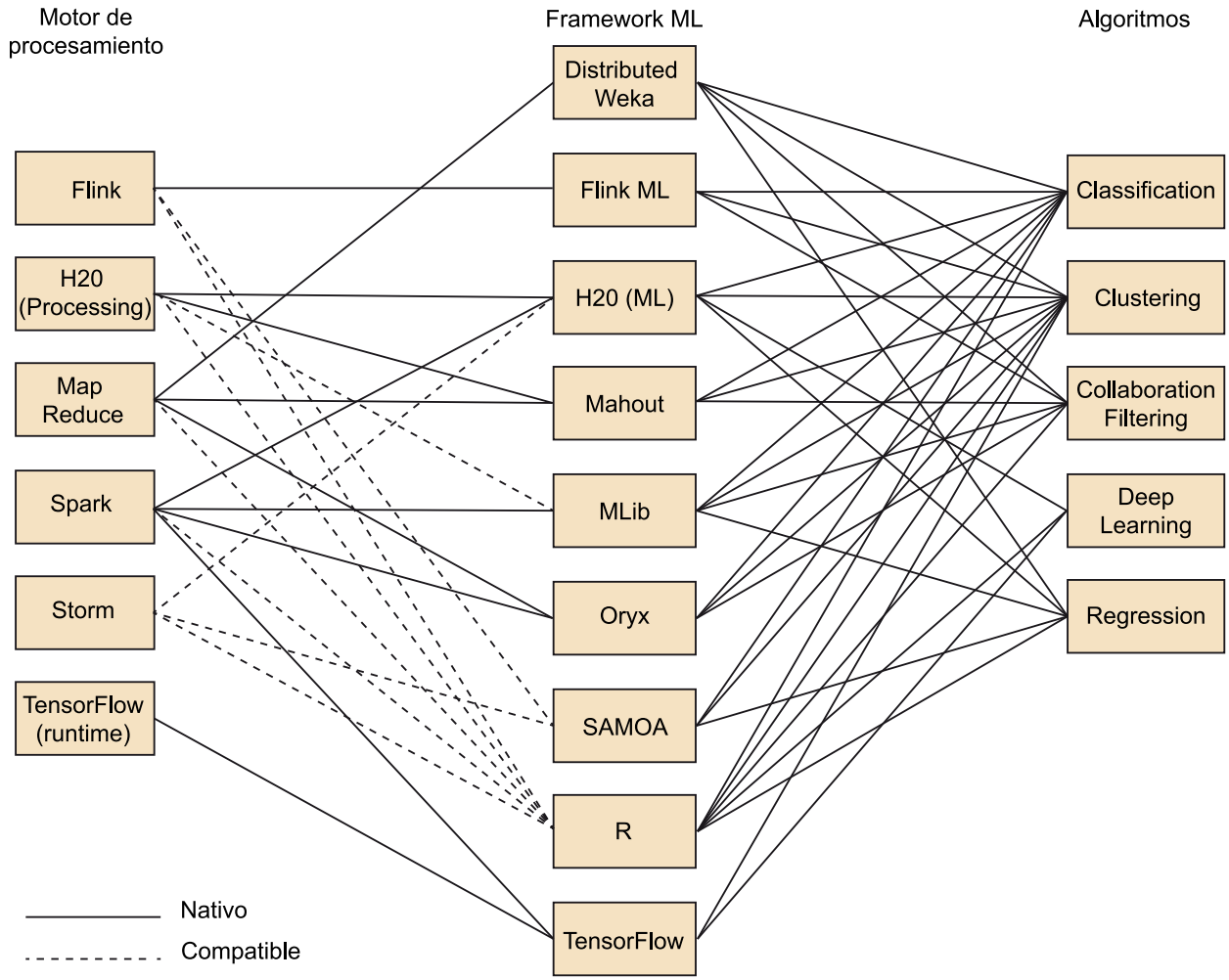
Uno de los aspectos más interesantes, pero que al mismo tiempo complica la elección y la implementación, es que las distintas librerías existentes pueden trabajar con diferentes motores de procesamiento. Recordemos que el motor de procesamiento se encarga de tareas de extracción, movimiento y manipulación del dato, mientras que los *framework* (o librería) puede entenderse como un contenedor de algoritmos de *machine learning* adaptados a *big data*.

Lectura complementaria

Landset y otros (2015). «A survey of open source tools for machine learning with big data in the Hadoop ecosystem». *Journal of Big Data* (2:24).

El soporte a los diferentes motores de procesamiento puede ser nativo o simplemente compatible mediante un conector. En la figura 6 presentamos cómo se relacionan algunas de las principales librerías presentadas con los motores de procesamiento y con la categoría de algoritmos soportado. Este último punto es relevante porque puede ayudarnos a decantarnos por una o por otra.

Figura 6. Comparativa librerías ML



Fuente: Josep Curto, ampliando el trabajo de Landset y otros.

3.4. Visualización

Tradicionalmente los componentes de inteligencia de negocio como los cuadros de mando, informes y/o vistas OLAP se han usado para presentar el resultado del análisis de la información. Con el advenimiento de *big data* y la combinación de tecnologías, este enfoque ya no es suficiente. Dos disciplinas han emergido para ayudar en la visualización de la información: *data visualization* (visualización de datos) y *data storytelling* (historias fundamentadas en datos). Debemos comprender primero estos conceptos.

> «Se entiende por *data visualization* la representación de datos que explota las habilidades visuales para amplificar los procesos cognitivos».

Data Visualization persigue incrementar las capacidades exploratorias y explicativas, representar grandes volúmenes de datos y comprender las relaciones ocultas en los datos de forma visual. Ha aparecido una gran colección de librerías especializadas en este ámbito. Entre estas librerías y herramientas destacan D3.js, Polimaps, Processing.js, Grafana, Tableau, Qliksense, CartoDB, Datawatch o Yellowfin.

Por otro lado,

> «Se entiende por *data storytelling* el método visual de presentar información para hacerla más comprensible y fácil de comprender».

En la actualidad, algunas herramientas propietarias como las que ofrecen Tableau, Qliksense, Quadrigram, Miso, TimelineJS o Yellowfin capacitan a las organizaciones para el uso de *data storytelling*, si bien también es posible crear de forma programática.

Lectura complementaria

S. Few (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.

Estas técnicas no solo tratan de usar la mejor representación para explicar lo que sucede, sino que además deben poderse conectar con los componentes de procesamiento y almacenamiento de *big data*. No solo se trata de tener la tecnología adecuada (escalable y adaptable a *big data*), sino de dominar la comunicación de la información. Como comenta Stephen Few, las capacidades para mostrar y explicar la información de manera efectiva no son intuitivas y es necesario aprender unos nuevos principios:

- Conocer la audiencia de la visualización. Esto incluye factores como el rol, el flujo de trabajo, el conocimiento técnico y de negocio de la audiencia.

- Determinar el valor que se quiere proporcionar a la audiencia. En este sentido, tenemos dos grandes opciones. Tenemos ya una pregunta que responder o estamos haciendo un análisis exploratorio. Esto incluye identificar lo que es relevante, estableciendo metas y expectativas.

- Seleccionar la visualización correcta. Esto incluye desde la elección del gráfico y/o la representación, al alcance, el horizonte temporal y el tipo de decisiones.

- Elegir las medidas adecuadas, que deben siempre ayudar a tomar decisiones.

- Creación/composición de la visualización, que debe tener en cuenta la forma, la estructura, la funcionalidad y los principios de diseño.

- Uso de criterios de diseño y presentación de información, como la elección de colores y tipografía.

Lectura complementaria

E. Segel; J. Heer (2010). *Narrative Visualization: Telling Stories with Data*. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis).

3.5. Sistemas híbridos

Tras discutir los sistemas de almacenamiento y el procesamiento en el contexto de *big data*, todo parece indicar que tenemos dos tipos de plataformas diferentes. Las vinculadas con el procesamiento *batch* y las de *streaming*. Este pensamiento no anda lejos de la realidad. Las necesidades y el diseño de estos sistemas son diferentes y, sin embargo, una organización puede necesitar ambos enfoques o incluso otros vinculados a NoSQL.

No es extraño que haya emergido una propuesta para una arquitectura de procesamiento de datos genérica, escalable y tolerante a fallos que sirva a ambos propósitos (*batch* y *streaming*). Esto es lo que conocemos como arquitectura Lambda, inventada por Nathan Marz.

Desde una perspectiva de alto nivel, esta arquitectura proporciona la siguiente funcionalidad:

- Los datos se distribuyen tanto a la capa *batch* como a la capa velocidad (en referencia a *streaming*) para su procesado.

- La capa de procesamiento *batch* tiene principalmente dos funciones:
  - Gestionar un conjunto de datos maestros a los que se añade el dato nuevo en bruto.

  - Precalcular las vistas *batch*.

- La capa de distribución indexa las vistas *batch* para que puedan ser consultadas en latencia bajas de forma *ad hoc*.

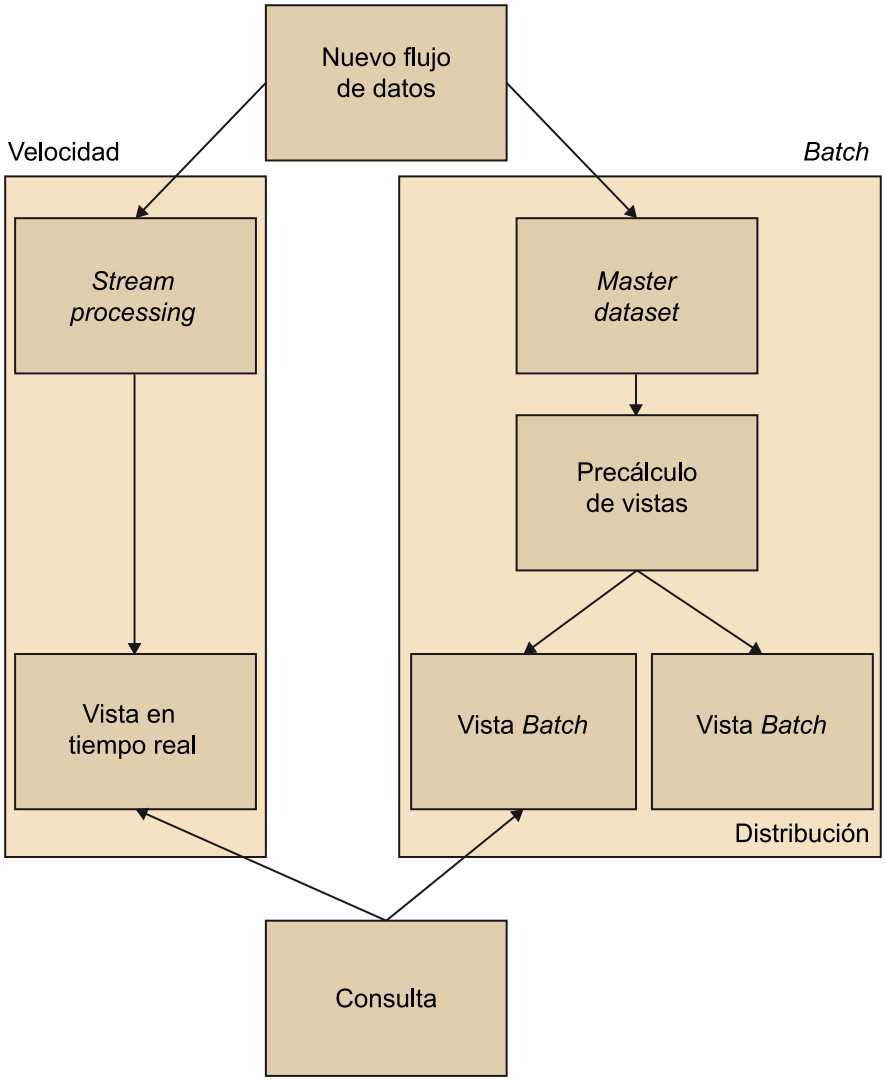
- La capa de velocidad complementa la capa de distribución proporcionando una mayor frecuencia de actualizaciones y solo trabaja con los datos más recientes.

- El resultado de una consulta puede ser el resultado de la combinación de datos en movimiento o en reposo.

Esta propuesta propone tener una arquitectura con los flujos separados en diferentes componentes. Por ejemplo, este tipo de sistema distribuido puede desplegarse en Apache Storm, creada por Nathan Marz y posteriormente cedida a la Apache Foundation, o en Apache Samza, basada en Apache Kafka, entre otros.

El funcionamiento de la arquitectura Lambda se ilustra en la figura 7.

Figura 7. Arquitectura Lambda



Fuente: Josep Curto, adaptado de Nathan Marz.

Este enfoque tiene el beneficio de que nos permite trabajar con ambos tipos de necesidades (en lotes y en flujo), pero tiene algunas desventajas, como que se combinan diferentes tecnologías de programación, lo que complica el diseño y el mantenimiento del código.

La propuesta de Marz no es la única en esta dirección. Existe otro enfoque que se fundamenta en el procesamiento *streaming*, la capacidad de procesamiento en paralelo y considerar que el procesamiento *batch* puede ser un caso particular de procesamiento *streaming*. Esta es la visión de Jay Kreps y se conoce como arquitectura Kappa.

Lectura complementaria

Para leer sobre la discusión original de la arquitectura Lambda, recomendamos el siguiente enlace: http://www.oreilly.com/ideas/questioning-the-lambda-architecture.

Desde una perspectiva de alto nivel, esta arquitectura proporciona la siguiente funcionalidad:

- El motor de procesamiento accede tanto a datos que provienen de un flujo continuo como a datos que están almacenados en un repositorio (con independencia del tipo de repositorio).

- Los datos se distribuyen tanto a la capa *batch* como a la capa velocidad (en referencia a *streaming*) para su procesado.

- La capa de procesamiento tiene principalmente dos funciones:
  - Gestionar la entrada y salida de datos, ya sea en movimiento o en reposo.
  - Poder responder/servir vistas históricas y/o en tiempo real.

- El resultado de una consulta puede ser el resultado de la combinación de datos en movimiento o en reposo.

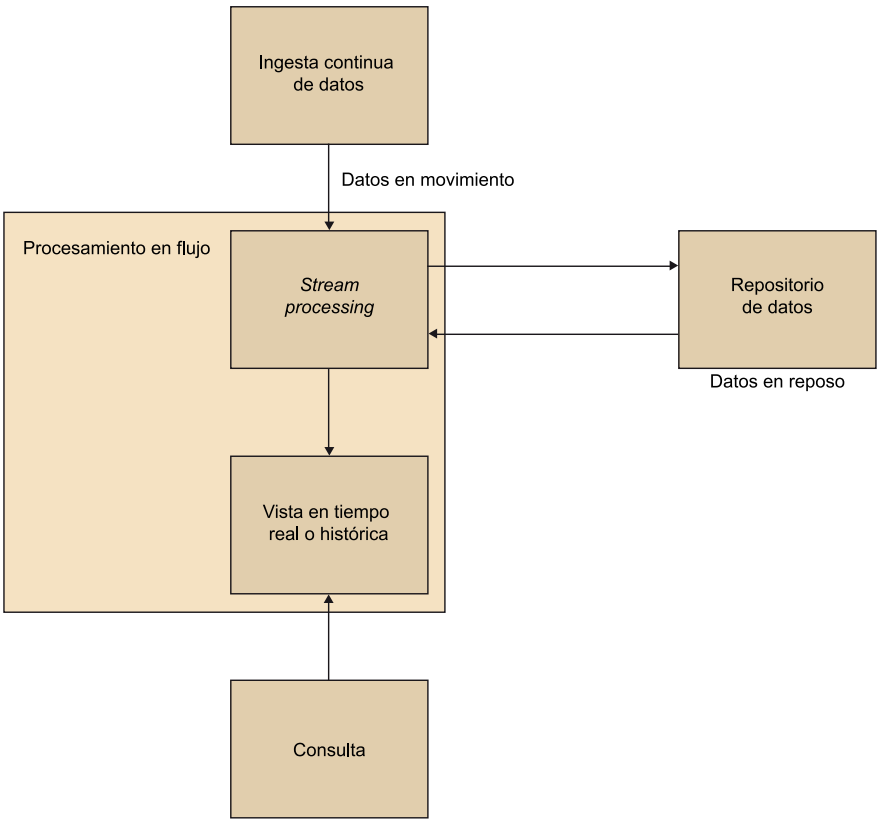
- En este enfoque se optimizan los componentes necesarios.

Por ejemplo, este tipo de sistema distribuido puede crearse usando Apache Flink, aunque puede lograrse con otros componentes, como Kafka, Samza o incluso Spark Streaming.

Este enfoque tiene el beneficio de que reduce la complejidad de mantenimiento del código, por lo que poco a poco va convirtiéndose en la opción preferida al compararla con la arquitectura Lambda.

El funcionamiento de la arquitectura Kappa se ilustra en la figura 8.

Figura 8. Arquitectura Kappa



Fuente: Josep Curto, adaptado de Jay Kreps.

Existen diferentes escenarios que puede necesitar el despliegue de una arquitectura Lambda o Kappa. Discutimos uno de ellos.

Cuando una empresa se enfrenta al fraude, como puede ser en el sector de las finanzas, energético o *retail*, tiene varias necesidades. Por un lado, necesita hacer un análisis forense de todo el histórico de transacciones para poder detectar nuevos patrones de fraude. Esta necesidad puede considerarse como un problema en el que prima la capacidad de trabajar con todo el historial y no la velocidad. Estamos ante una necesidad que puede cubrirse con el procesamiento y almacenamiento *batch*.

Por otro lado, también existe otra necesidad una vez se han reconocido los patrones, que consiste en analizar el flujo de transacciones en tiempo real y detectar si se cumple alguno de los patrones. Aquí prima la velocidad de detección del evento. Estamos ante un escenario de procesamiento en *streaming*.

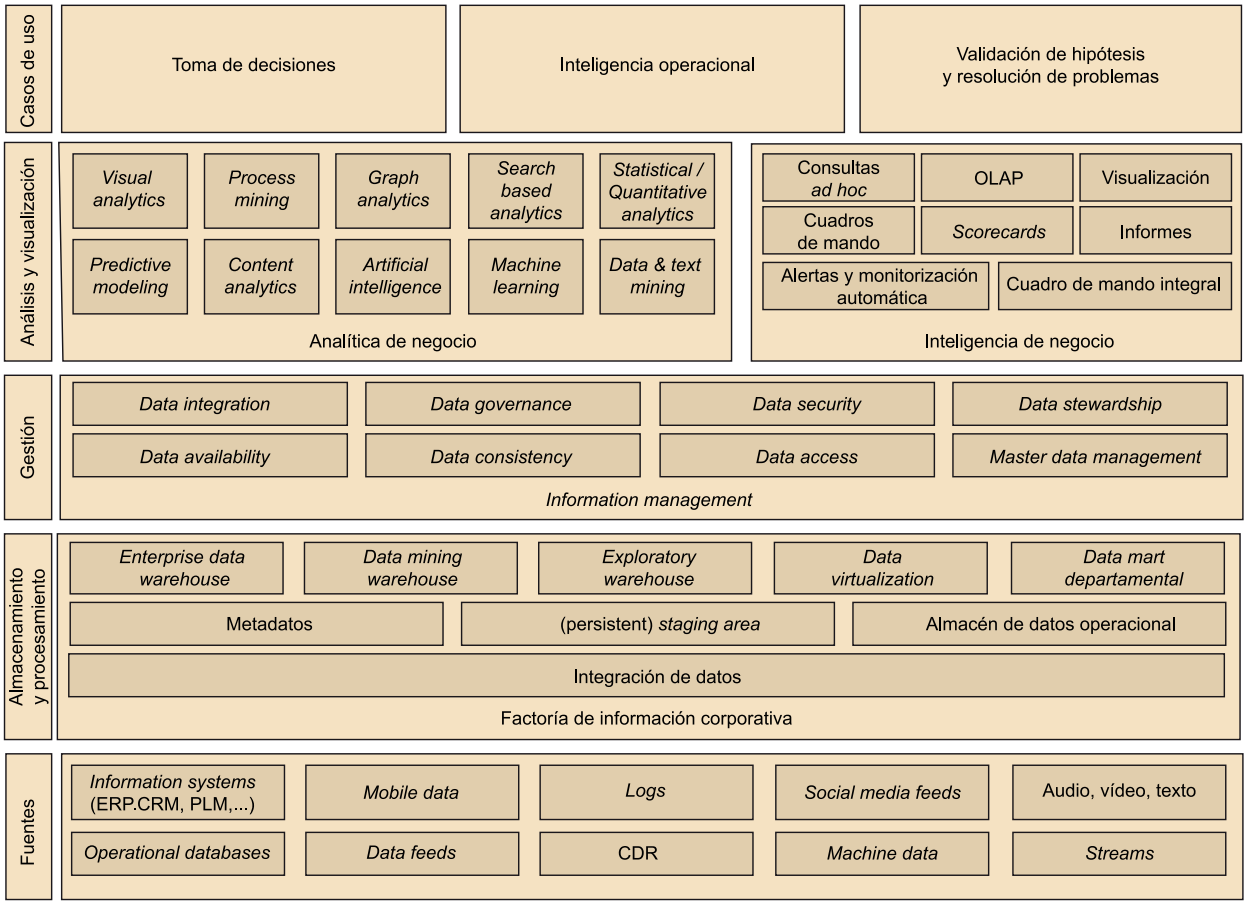
4. Arquitectura y ecosistemas de *big data*

Anteriormente hemos introducido las diferentes tecnologías existentes. A saber, almacenamiento, procesamiento, análisis y visualización. En este capítulo nos centraremos en cómo se agrupan estas tecnologías. Por un lado, formando una arquitectura empresarial de datos y, por otro, en ecosistemas de componentes integrados.

4.1. Arquitectura de *big data*

Como ya hemos comentado, antes de la emergencia de *big data* las organizaciones ya analizaban los datos para generar valor. La arquitectura de datos presentaba diferentes capas y componentes, como se ilustran en la figura 9.

Figura 9. Arquitectura genérica de datos



Fuente: Josep Curto.

Hemos introducido ya los componentes de la capa de análisis y los elementos de la capa de uso.

En la capa de almacenamiento y procesamiento, tenemos diferentes elementos. Como ya hemos visto, el *data warehouse* es el repositorio de la información relevante para la toma de decisiones; los metadatos, que en este caso están vinculados a esta capa y a su gestión, catalogación y explotación; y la integración de datos, que permite la captura, el procesamiento y la distribución del dato. Tenemos también otros elementos, como el *data mart*, que es un subconjunto del almacén de datos, almacenes de datos especializados como el dedicado a la minería de datos, a la exploración de datos, el operacional para dar respuestas intradía o la *staging area*, que se usa frecuentemente como caché de datos no persistente.

Lectura complementaria

J. Conesa; J. Curto (2012). *Introducción al Business Intelligence*. Barcelona: Editorial UOC.

En la capa de gestión tenemos varios componentes, que se encargan de la disponibilidad, de la gobernanza, de la consistencia, de la seguridad, del acceso, de la gestión de datos maestros, de la administración e incluso de la integración del dato.

Las diferentes capas presentadas pueden estar combinadas en una única plataforma o pueden estar formadas por diferentes componentes que se deben integrar para que trabajen de manera unísona. Cabe destacar que a pesar de que la capa de gestión es una de las más importantes, frecuentemente se ve reducida a la integración del dato e incluso solo como el componente de procesamiento de la factoría de información.

A medida que crece la complejidad en una organización, la capa de gestión adquiere más y más importancia, lo que culmina con el despliegue de programas de gobernanza del dato.

La arquitectura de datos actuales bebe de la anterior pero debe introducir el soporte para *big data* y cada uno de sus componentes está evolucionando a una gran velocidad por diversos motivos:

- Para ampliar sus capacidades respecto a fuentes de datos. Por ejemplo, extendiendo la cantidad de conectores disponibles, que van desde nuevas bases de datos hasta conectividad con sensores y dispositivos incluso en escenarios de *edge analytics*.

Edge analytics

Cuando hablamos de *edge analytics* hacemos referencia a aplicaciones analíticas para IoT en las que ciertos algoritmos se ejecutan en los nodos de la red y no solo en el centro de datos.

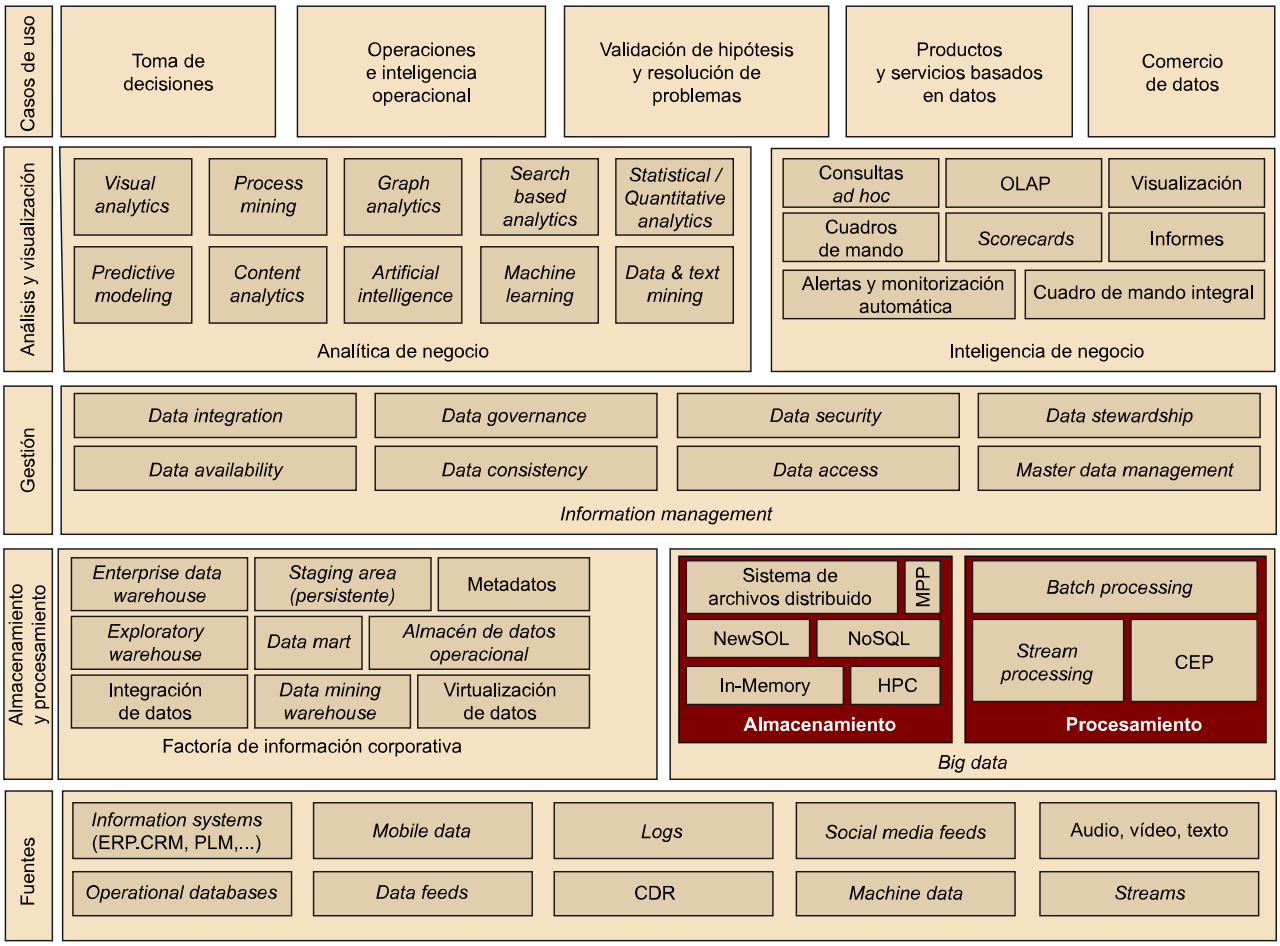
- Para proporcionar flexibilidad en el uso de componentes permitiendo combinar aquellos que son relevantes para el caso de negocio que se está resolviendo. Por ejemplo, capacidad de usar Apache Hadoop, Spark o Flink como motor de ejecución.

- Para proporcionar una arquitectura empresarial, poniendo foco en la seguridad.

- Para extender la capacidad de gobernanza a conjuntos de datos complejos, lo que supone redefinir las políticas de catalogación e indexación semántica de datos.

Esto al final se traduce en evolución de la capa de procesamiento y almacenamiento, pero que –como hemos comentado en el capítulo anterior– afecta a cada una de las capas de la arquitectura del dato. La figura 10 representa un esquema genérico de arquitectura para *big data* que –como es posible comprobar– amplía la arquitectura anterior.

Figura 10. Arquitectura genérica de *big data*



Fuente: Josep Curto.

Es posible apreciar que las tecnologías de *big data* extienden la arquitectura de datos incrementando la capacidad de generar valor en una organización a través de nuevos casos de uso.

Hay escenarios en los que el objetivo es crear una aplicación de negocio fundamentada en necesidades muy específicas que no encajan en una plataforma de un fabricante o incluso en un único ecosistema. En tal caso, la propuesta del ingeniero del dato es posible que no se fundamente en una plataforma empresarial de *big data*, sino que esté conformada por una serie de componentes que integrar o incluso que desarrollar.

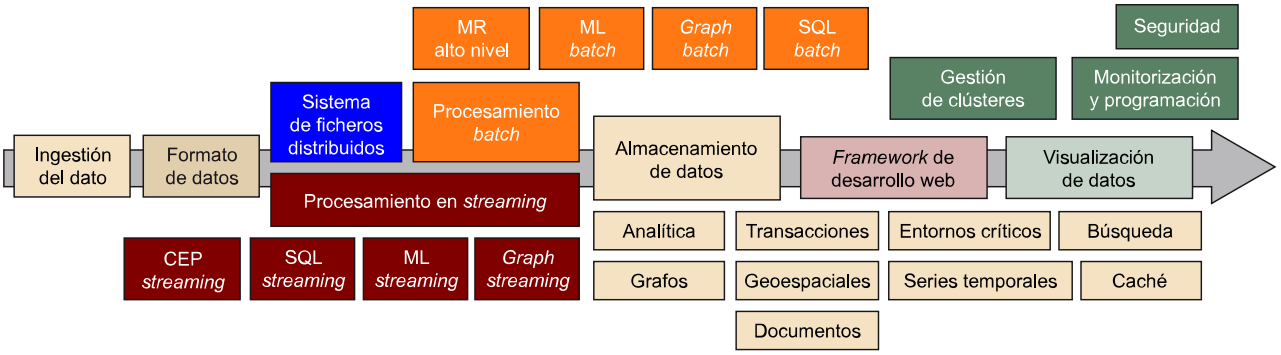
Es decir, se observa una evolución de los ecosistemas de *big data* en dos direcciones:

- **Soluciones *enterprise-ready***: son plataformas que cumplen estándares de integración y seguridad, y que permiten responder a una serie de casos de uso específicos.

- **Soluciones *ad hoc***: plataformas basadas en componentes independientes que es necesario integrar internamente. Es decir, hay diferentes módulos para elegir y frecuentemente cada uno de ellos es una tecnología *best of breed*. Este segundo escenario está muy ligado al desarrollo de productos y servicios basados en datos. En este punto es donde el ingeniero del dato desempeña un papel fundamental.

La figura 11 representa un arquitectura *ad hoc* a través del flujo de datos, desde su extracción hasta la presentación de resultados.

Figura 11. Arquitectura *ad hoc* de *big data*



Fuente: Josep Curto, ampliando el trabajo de Insight Data Engineering.

En la creación de un producto o servicios de datos, será necesario tener muy claro las necesidades que tiene la organización y seleccionar la tecnología adecuada. Tenemos los siguientes componentes:

- **Ingestión del dato**: componente que habilita la captura de datos. Ejemplos: Kafka, Logstash, RabbitMQ, Fluentd, Chuckwa y AWS Kinesis.

- **Formato de datos**: componente que habilita la transformación del dato a formatos más optimizados para su almacenamiento en bruto, como el formato binario. Ejemplos: Avro, ProtoBuf, Thrift y Parquet.

- **Sistema de archivos distribuido**: componente que habilita el almacenamiento de los datos en bruto en un sistema de archivos distribuidos. Ejemplos: HDFS, AWS S3, Microsoft Azure, Alluxio y Ceph.

- **Procesamiento *batch***: componente que habilita el procesamiento *batch*. Ejemplos: Spark, Hadoop MapReduce, AWS EMR, Flink y Tez.

  - **MR alto nivel**: componente que habilita el uso de MapReduce mediante *scripting*. Ejemplos: Pig, Cascading, Hadoop Streaming y Cascalog.

  - **ML *batch***: componente que habilita *machine learning* en procesamiento *batch*. Ejemplos: Mahout, Spark Mlib, FlinkML y H_20.

  - ***Graph batch***: componente que habilita el análisis de grafos en procesamiento *batch*. Ejemplos: GraphLab, Giraph, Spark GraphX y Hama.

  - **SQL *batch***: componente que habilita el uso de SQL en procesamiento *batch*. Ejemplos: Hive, Presto, Drill, Hue e Impala.

- **Procesamiento en *streaming***: componente que habilita el procesamiento en *streaming*. Ejemplos: Storm, Spark Streaming, AWS Lambda, Samza, Akka y Flink.
  - **ML *streaming***: componente que habilita *machine learning* en procesamiento en *streaming*. Ejemplos: Spark Mlib y SAMOA.

  - **Graph *streaming***: componente que habilita el análisis de grafos en procesamiento en *streaming*. Ejemplos: X-stream/Chaos.

  - **SQL *streaming***: componente que habilita el uso de SQL en procesamiento en *streaming*. Ejemplos: Spark SQL y Flink Table API.

  - **CEP *Streaming***: componente que habilita CEP en procesamiento en *streaming*. Ejemplo: Flink CEP.

- **Almacenamiento de datos**: componente que habilita el almacenamiento de datos. Existen diferentes bases de datos específicas: analíticas, de grafos, transaccionales, geoespacial, de entornos críticos, de series temporales, de búsqueda y de caché.
  - **Analítica**: componente de almacenamiento de datos que habilita la ejecución nativa del algoritmos y técnicas analíticas. Ejemplos: AWS Redshift, Teradata y Vertica.

  - **Grafos**: componente de almacenamiento de datos que guarda la información en forma de grafo y habilita su análisis. Ejemplos: Neo4j, OrientDB y ArangoDB.

  - **Transaccionales**: componente de almacenamiento de datos optimizada para transacciones que soportan ACID. Ejemplos: MySQL, Oracle, Microsoft SQLServer y PostgreSQL.

  - **Geoespacial**: componente de almacenamiento de datos optimizada para datos geolocalizados. Ejemplos: PostGIS y Elasticsearch.

  - **Documentos**: componente de almacenamiento de datos optimizada para documentos. Ejemplo: MongoDB y CouchDB.

  - **Entornos críticos**: componente de almacenamiento de datos optimizada para entornos críticos. Ejemplos: Cassandra, Riak y AWS DynamoDB.

  - **Series temporales**: componente que habilita el almacenamiento y el análisis de datos como una serie temporal. Ejemplos: InfluxDB, Cassandra y Druid.

  - **Búsqueda**: componente de almacenamiento de datos optimizada para la búsqueda de información. Ejemplos: ElasticSearch, Solr y MongoDB.

  - **Caché**: componente de almacenamiento de datos que usa *in-memory* para acelerar consultas frecuentes. Ejemplos: Redis, Memcache y Hazelcast.

- ***Framework* de desarrollo web**: componente que habilita el desarrollo de un entorno web. Ejemplos: Ruby on Rails, Node,js, Django, AngularJS y Flask.

- **Visualización de datos**: componente que habilita la visualización del dato. Ejemplo: D3,js, Tableau, Qliksense, Leaflet, Highcharts y Kibana.

- **Gestión**: componentes principalmente usados en producción como gestión de clústeres, monitorización/programación y seguridad.
  - **Gestión de clústeres (y recursos)**: componente que se encarga de la gestión eficiente de clústeres. Ejemplos: Docker, Zookeeper, YARN, Mesos, REEF y Helix.

  - **Monitorización/programación**: componente que se encarga de la monitorización de los diferentes componentes y de la programación de tareas. Ejemplos: Luigi, Airflow, Nagios, Graphite y Azkaban.

  - **Seguridad**: componente que se encarga de la seguridad del dato. Ejemplos: Sentry, RecordService y Knox.

Para ejemplificar la arquitectura anterior, vamos a discutir un caso.

Imaginemos que el ayuntamiento de una ciudad española tiene una red de dispositivos medioambientales con múltiples sensores desplegados en varios puntos con el objetivo de medir y comprender la evolución mensual de la contaminación.

Los datos en bruto procedentes de los sensores son capturados a través del componente de ingestión del dato de manera continua en tiempo real. En la capa de formato de datos se les aplican transformaciones para homogeneizar las unidades. Aunque la organización quiere solo el análisis mensual, reconoce que en el futuro podría estar interesada en otro tipo de análisis y considera dos tipos de datos. Por un lado, en bruto dentro de un sistema de ficheros distribuidos y, por otro, especializado en serie temporal mensual. Por tanto, primero se guarda el dato en el sistema de ficheros y después mediante la capa de procesamiento *batch* se consolida y se generan los indicadores de contaminación, y se guarda en la base de datos especializada.

Posteriormente, a través del componente de visualización, se accede a un cuadro de mando resumen de la evolución de los principales indicadores tanto en gráficas como en mapas.

4.2. *Data lakes*

Un nuevo enfoque de arquitectura está emergiendo en el contexto de *big data*, denominado *data lake*. Debemos definir este concepto primero.

Se entiende por ***data lake*** el repositorio de información de una organización que incluye tanto los datos estructurados como no estructurados consolidados en una única tabla.

Es necesario comentar que no se trata simplemente de almacenar todos los datos de una organización, sino de establecer un proceso sistemático para que este almacenamiento pueda ser explotado.

En primera instancia, esto significa que el éxito del *data lake* está ligado a un programa de gobernanza del dato que incluya catálogo, auditoría, trazabilidad, control de calidad, control de acceso y gestión de metadatos. Construir un catálogo de datos eficiente, correcto y comprensivo requiere combinar múltiples enfoques, como *machine learning*, *natural language processing* (NPL), técnicas de inferencia estadística, indexación e identificación automática de datos y recursos, y la participación de los expertos en la organización. Estamos hablando de una combinación entre curación automática y experta del dato. Nuevas componentes de Hadoop como Apache Atlas o compañías como Alation buscan ayudar en este punto.

NPL

Cuando hablamos de NLP hacemos referencia a un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

En segundo lugar, es necesario cambiar el modo como una organización se acerca al *big data*. El punto de partida no debe ser la tecnología ni el dato, debe ser el negocio y esto se traduce en validar las oportunidades a través de tres preguntas:

- **Estratégica**: ¿Es una oportunidad estratégica para la organización?
- **Accionable**: ¿Puede el negocio apalancarse sobre el conocimiento generado?

- **Factible**: ¿Superan los beneficios a los costes?

Aunque en primera instancia uno pueda pensar que se trata de clarificar una serie de dicotomías, en realidad es necesario analizar cada oportunidad de una manera sistemática. Al final esta aproximación se traduce en que la construcción del *data lake* pasa a ser necesariamente un proceso incremental y no un *big bang*.

El concepto que hemos introducido en cierta medida guarda similitud con el *data warehouse* tal y como apuntan los detractores del concepto, pero va mucho más allá. En la tabla 6 se resumen las diferencias entre estos dos conceptos.

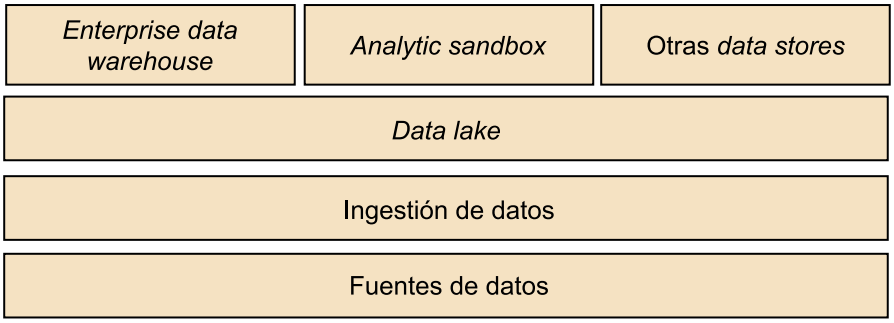
Tabla 6. Comparativa *data lake* frente a *data warehouse*

Factor	*Data warehouse*	*Data lake*
Cargas de trabajo	Centenares usuarios concurrentes hacen análisis de datos utilizando las capacidades de gestión de carga de trabajo para mejorar el rendimiento por consulta. Procesamiento por lotes.	El procesamiento por lotes de los datos a gran escala. Permite incrementar el número de usuarios.
Esquema	El esquema de datos se diseña de antemano antes del almacenamiento usando técnicas de modelización, lo que se conoce como *schema on write*.	El esquema de datos se diseña después del almacenamiento, lo que se conoce como *schema on read*.
Escala	Escala por coste moderado	Escala por bajo coste
Método de acceso	Basado en SQL y herramientas de BI	Basado en programas y sistemas SQL-like
Beneficios	Respuestas rápidas a consultas. Rendimiento consistente. Alta concurrencia. Fácil consumo. Visión estructurada de la empresa. Análisis cruzado. Se transforma el dato una vez, se consume varias.	Ejecución distribuida para escalabilidad. Paralelización de lenguajes de programación (Python, Java, etc.). Soporte a consultas SQL-like como Hive o Pig. Cambio de las economías de escala de almacenamiento de datos.
Soporte SQL	ANSI SQL, ACID	Basado en programación/Soporte lenguajes similares a SQL
Calidad del dato	Limpio	En bruto
Tipo de acceso	Basado en búsquedas	Basado en escaneos
Complejidad	En las consultas	En el procesamiento
Coste/Eficiencia	Eficiencia del uso de CPU/IO	Bajo coste de almacenamiento y procesamiento

La existencia del *data lake* no supone la desaparición del *data warehouse*, sino la creación de una arquitectura más compleja en la que se combinan ambos elementos. Una forma de pensar es que el *data lake* supone una fase anterior al *data warehouse* y al entorno analítico. Además, esta arquitectura no cubre actualmente escenarios de procesamientos en *streaming*, lo que nos indica que

actualmente solo cubre una parte de los sistemas híbridos presentados en el subapartado 3.5. La figura 12 presenta el *data lake* dentro de la arquitectura de datos.

Figura 12. Arquitectura genérica de data lake



Fuente: Josep Curto.

Por tanto, el *data lake* se transforma en la fuente de datos del *data warehouse*, de bases de datos analíticas y del resto de las bases de datos existentes para las estrategias de datos. En cierto modo puede considerarse como un almacén de datos operacional persistente.

La creación de repositorios únicos como los comentados es deseable, pero están supeditados al historial de la organización, a sus necesidades, a su tamaño y a las regulaciones del sector en el que opera, que pueden limitar su efectividad y sus beneficios. Esto se traduce en que la arquitectura será más complicada y en que el *data lake* solo podrá dar servicio a una región específica y a ciertos casos de uso. Por lo tanto, una organización puede llegar a tener más de uno y en combinación con otros sistemas.

4.3. Ecosistemas

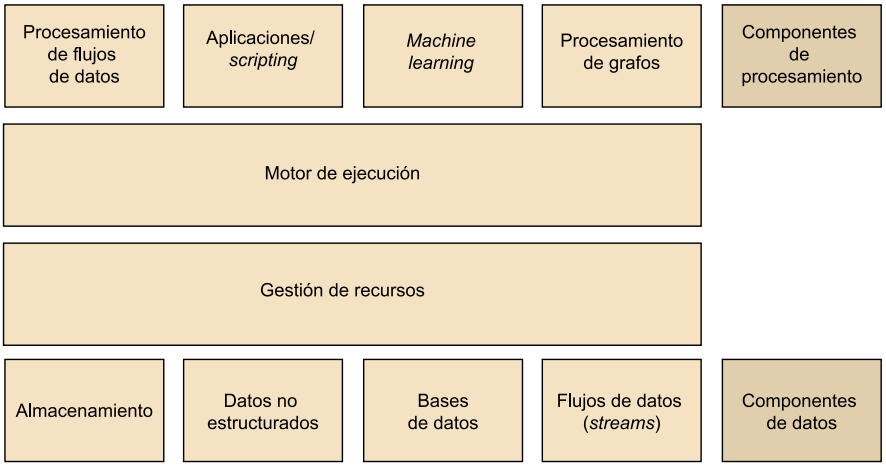
Las tecnologías de *big data* se suelen estructurar en torno a ecosistemas de software integrados. Primero debemos entender qué significa este concepto.

> «Se entiende por ecosistema software el espacio de trabajo en el que conviven una serie de herramientas que acompañadas de unas buenas prácticas permiten a un equipo de desarrollo modelar una metodología de trabajo».
>
> Brett Proter

Esto se traduce en una colección de componentes tecnológicos integrados para resolver un problema específico de negocio. Como hemos discutido, las tecnologías de *big data* buscan ayudar a la generación de valor de datos complejos y existen diferentes tipos, como se ilustra en la tabla 2. El ecosistema puede entenderse como la implementación de una determinada arquitectura ya sea para el procesamiento *batch*, en *streaming* o ambas.

En general, un ecosistema de tecnologías *big data* incluye los siguientes componentes, como se ilustra en la figura 13.

Figura 13. Ecosistema genérico



Fuente: Josep Curto.

Explicamos en qué consiste cada uno de los componentes:

- **Componentes de datos**: consiste en una serie de conectores que permite el acceso a fuentes diferentes de datos (bases de datos, flujos de datos o datos no estructurados), así como a diferentes sistemas de almacenamiento.

- **Componente de gestión de recursos**: se encarga de gestionar las diferentes tareas que realiza la plataforma (por ejemplo, trabajos de procesamiento) y componentes del sistema (clúster y nodos).

- **Componente de motor de ejecución**: se encarga ejecutar los diferentes trabajos de procesamiento y almacenamiento iniciados en el sistema.

- **Componente de procesamiento**: consiste en una serie de componentes que permiten el procesamiento del dato en función de la necesidad de negocio y su forma de tratamiento (en flujo, grafos, consultas o aplicaciones y *machine learning*).

Es fácil detectar una analogía entre este esquema y la arquitectura *ad hoc* en la que se desglosan más cada uno de los componentes.

A medida que madura un ecosistema, la cantidad de componentes va creciendo con el objetivo de proporcionar una solución que ofrezca prestaciones empresariales y cubra múltiples escenarios, lo que se traduce en nuevos componentes, como:

- **Configuración**: que permite la parametrización de la configuración del ecosistema.

- **Seguridad**: que permite gestionar la seguridad del dato y del sistema.

- **Aprovisionamiento**: que permite ajustar de manera manual, semiautomática y/o automática las necesidades de arquitectura de hardware del sistema.

En los siguientes subapartados vamos a revisar la situación actual de algunos de los principales ecosistemas de *big data*. Es importante comentar que los ecosistemas están en constante evolución, por lo que pueden aparecer nuevos componentes en cada una de las capas.

Además de los que se van a explicar, existen los ecosistemas de proveedores de *big data* en modalidad *cloud computing*, ya sea IaaS (infraestructura como servicio), Paas (plataforma como servicio) o SaaS (software como servicio), entre los que destacan Google, Microsoft o Amazon.

También está emergiendo un nuevo tipo de ecosistemas basado en la tecnología docker, como, por ejemplo, Pachyderm o la plataforma propuesta por el proyecto Big Data Europe auspiciado por la Comisión Europea.

Por último, han emergido ecosistemas no fundamentados en una tecnología central, como:

- **SMACK**: hace referencia a una combinación de componentes en la que se considera que cada dato es un evento. Este ecosistema incluye Spark (*data processing*), Mesos (*cluster resource management*), Akka (*message-driven app toolkit*), Cassandra (*storage engine*) y Kafka (*event-processing framework*).

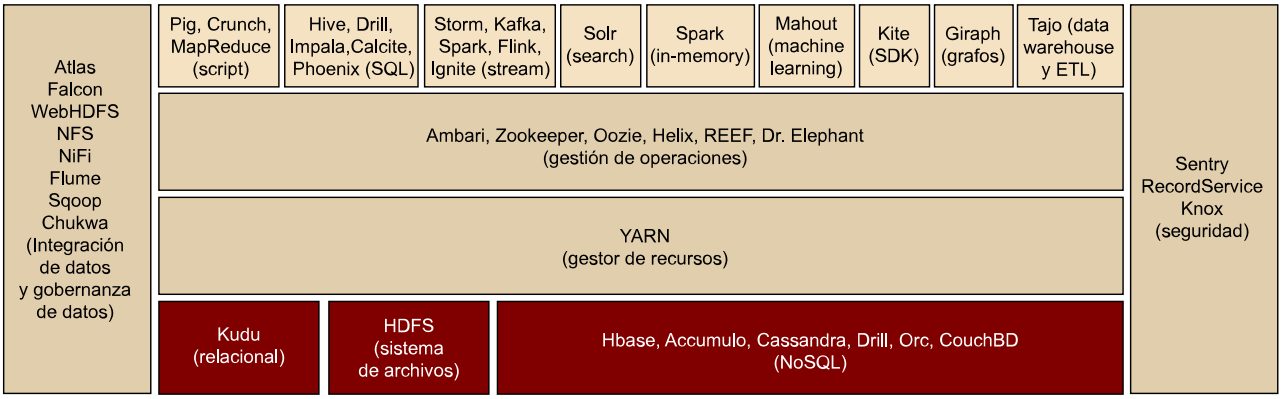
- **PANCAKE STACK**: hace referencia a una combinación de componentes en la que el foco es un sistema de recomendaciones. Este ecosistema incluye componentes como Presto, Arrow, Nifi, Cassandra, Airflow, Kafka, ElasticSearch, Spark, Tensorflow, Algebird, CoreNLP y Kibana.

4.3.1. Ecosistema Apache Hadoop

Apache Hadoop es una plataforma *open source* escrita en Java para el procesamiento y almacenamiento de datos distribuidos de grandes volúmenes de datos sobre clústeres de servidores. Su origen podemos encontrarlo en 2003 en el artículo «The Google File System», escrito por Sanjay Ghemawat, Howard Gobioff y Shun-Tak Leung, y actualmente ha evolucionado hacia un ecosistema que va más allá de los componentes de almacenamiento y procesamiento. Hadoop está optimizado para escenarios de *batch processing*.

El ecosistema de Apache Hadoop se estructura como se ilustra en la figura 14.

Figura 14. Ecosistema Apache Hadoop



Fuente: Apache Hadoop.

Este ecosistema incluye múltiples componentes:

- Para la ingestión del datos y gobernanza de datos, como NiFi, Falcon para la gobernanza de datos o Sqoop para mover datos entre HDFS y bases de datos relacionales.
- Para el almacenamiento de datos relacionales, como Kudu.
- Para sistemas de archivos distribuidos, como HDFS.
- Para NoSQL: Accumulo, Cassandra, Hbase, Orc o CouchDB.
- Para la seguridad, como Sentry o Knox.
- Para la gestión de recursos, como YARN.
- Para la gestión de operaciones: Ambari, Oozie, Zookeper o Dr. Elephant.
- Para el procesamiento de datos de diferentes formas:
  - *Script*: Pig, MapReduce o Crunch para la creación y validación de procedimientos complejos para MapReduce.
  - SQL: Hive, Phoenix, Impala, Drill o Calcite.
  - *Stream*: Storm, Kafka, Spark, Flink, Ignite.
  - *Search* (búsqueda): Solr.
  - *Machine learning*: Mahout.
  - SDK: Kite.
  - Grafos: Giraph.
  - *Data warehouse*: Tajo.

Una de las particularidades del ecosistema de Hadoop es la posibilidad de usar diferentes motores de procesamiento, como MapReduce, Spark o Flink.

Como se puede apreciar, Apache Hadoop es un ecosistema rico en componentes y que ya ha alcanzado una madurez considerable, pero sigue en evolución con múltiples proyectos en incubación. Existen también proyectos de la fundación Apache vinculados con *big data* pero no integrados todavía en el ecosistema de Hadoop, como Airvata, Arrow, Apex, Bigtop, Gora, Metamodel o VXquery. Estos componentes, y los que están en proceso de incubación, están llamados a redefinir de nuevo tanto el ecosistema como las plataformas del futuro.

Algunas de las combinaciones de componentes están disponibles solo en algunos fabricantes. Varios fabricantes tienen su propia plataforma integrada, como HortonWorks, Cloudera, MapR, Pivotal, IBM, SAP o Microsoft, entre otros.

4.3.2. Ecosistema Apache Spark

Apache Spark es una plataforma *open source* para el procesamiento de datos que nace para superar las limitaciones de MapReduce basado en el paradigma *resilient distributed datasets* (RDD). Estas limitaciones consisten en que MapReduce fuerza a seguir un proceso lineal de lectura, mapeo, reducción y almacenamiento del dato en disco, lo que supone unas latencias de escritura y lectura. Spark supera esta limitación a partir del uso de la memoria (distribuida) del clúster en lugar de la escritura a disco.

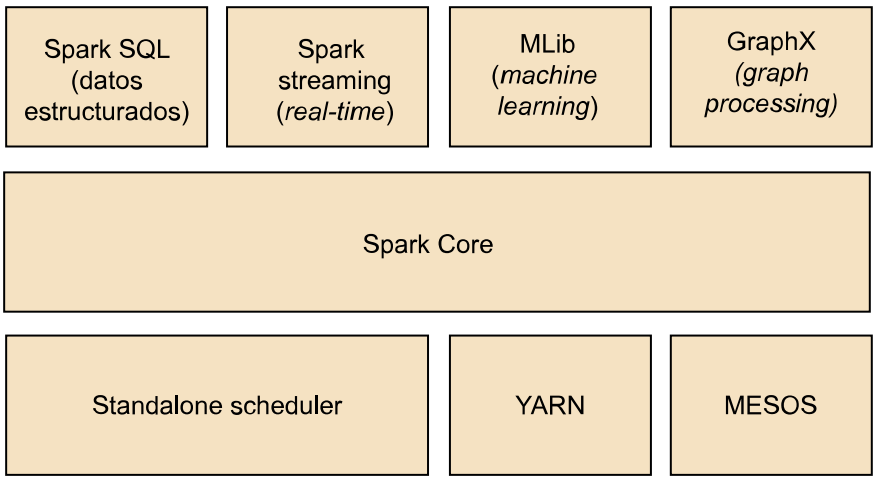
Lectura complementaria

M. Zaharia y otros (2012). *Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing*. Berkeley: University of California.

El origen de Spark podemos encontrarlo en los desarrollos de Matei Zaharia (actual CTO de Databricks) en 2009 cuando estaba en UC Berkeley AMPLab. En 2010 pasó a convertirse en *open source*. Spark soporta escenarios de *batch processing* y *streaming processing* por encima del segundo.

El ecosistema de Apache Spark se estructura como se ilustra en la figura 15.

Figura 15. Ecosistema Apache Spark



Fuente: Apache Spark.

El ecosistema de Spark incluye componentes para la gestión, como YARN y MESOS, un núcleo de procesamiento y varios componentes de procesamiento de datos estructurados, *streaming*, *machine learning* y grafos. La comunidad de Spark es bastante activa y está desarrollando múltiples conectores para extender el ecosistema presentado.

Databricks es la principal empresa detrás de este ecosistema, si bien muchos otros fabricantes están incluyendo Apache Spark como componentes de su plataforma, como HortonWorks, Cloudera, MapR, Qubole o Stratio, entre otros.

También hay empresas que ofrecen plataformas de análisis de datos fundamentadas en Apache Spark, como Arimo, Alpine Data, ClearStoryData, Sailthru o Zaloni.

4.3.3. Ecosistema Apache Flink

Apache Flink es una plataforma *open source* para el procesamiento de datos en modalidad *batch* y *streaming*. El enfoque es considerar que todo son flujos y que el procesamiento *batch* es un caso particular. Flink soporta escenarios que necesitan capturar y analizar datos por debajo del segundo.

Su origen podemos encontrarlo en 2010 en el proyecto de investigación colaborativo *Stratosphere: Information Management on the Cloud*, en el que participaron Technical University Berlin, Humboldt-Universität zu Berlin y Hasso-Plattner-Institut Potsdam. Es, por lo tanto, uno de los pocos proyectos europeos vinculados a esta área.

El ecosistema de Apache Flink se estructura en diferentes componentes:

- DataStream API, para procesamiento en *streaming*, con soporte para Java y Scala.
- DataSet API, para procesamiento *batch*, con soporte para Java, Scala y Python.

API (*application programming interface*)

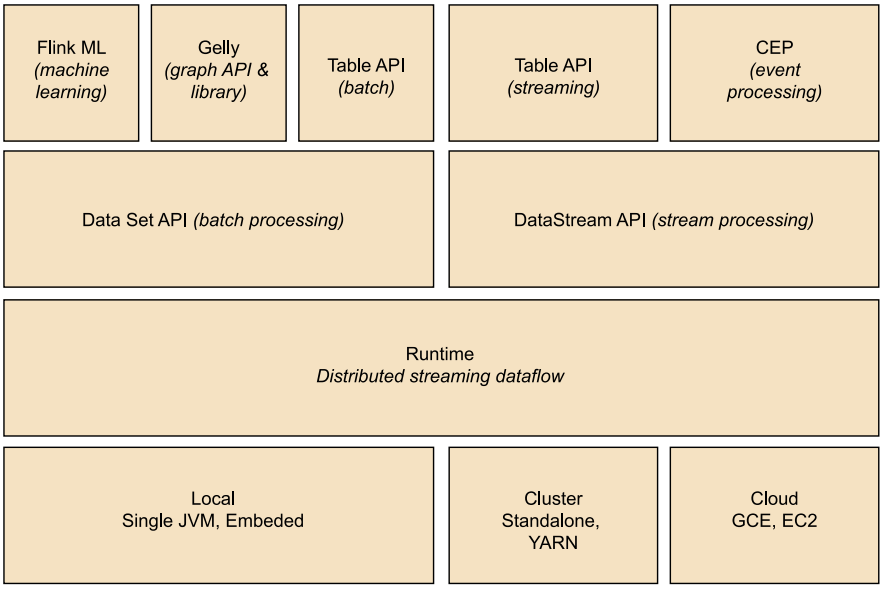
Cuando hablamos de API (*application programming interface*), o interfaz de programación de aplicaciones, hacemos referencia al conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Además, incluye librerías específicas para el análisis de datos, como:

- Table API, que soporta expresiones similares a SQL embebidas en Java y Scala.
- Flink ML, librería de *machine learning*.
- Gelly, librería de procesamiento de grafos.
- CEP, para el procesamiento de eventos.

El ecosistema de Apache Flink se estructura como se ilustra en la figura 16.

Figura 16. Ecosistema Apache Flink



Fuente: Apache Flink.

Apache Flink se puede desplegar de forma local, en un clúster o en la nube. DataArtisans es la empresa que da soporte a este ecosistema, si bien, como en el caso de Spark, otras empresas están integrando Flink, como MapR.

4.3.4. Ecosistema Apache Alluxio

Apache Alluxio es otro proyecto *open source* de UC Berkeley que surge para optimizar un ecosistema ya existente, en este caso, el de Spark. Tradicionalmente, la tolerancia a fallos se ha abordado mediante la replicación. Esto supone en una red de clústeres un uso intensivo de la red para realizar copias, lo que se convierte en una potencial limitación. Haoyuan Li, creador de Alluxio, propone una alternativa para minimizar la replicación consistente en crear un *log* con todos los cambios de cada registro de fácil acceso y solo una máquina del clúster trabaja. En el caso de error, Alluxio recupera los cambios del *log* y trabaja con otro servidor, lo que reduce la cantidad de datos en movimiento en el clúster y busca apalancarse en la potencia de procesamiento.

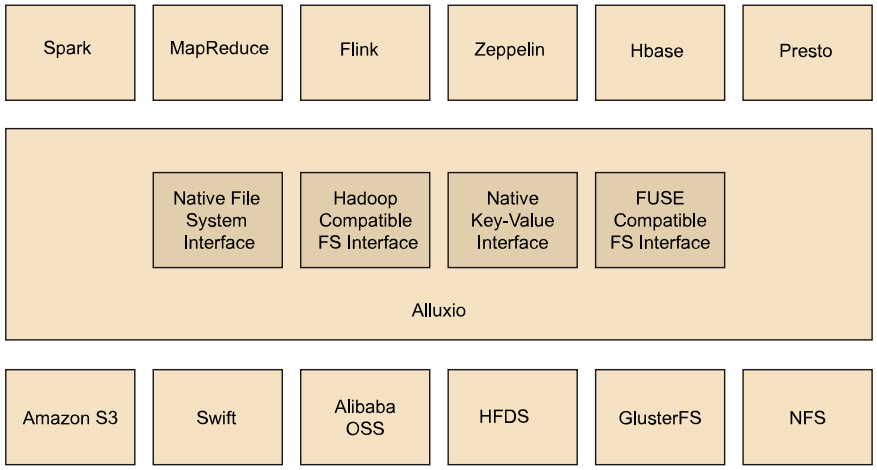
El ecosistema de Apache Alluxio se estructura en diferentes componentes:

- Conectores a diferentes sistemas de almacenamiento de datos, como Amazon S3 o HDFS.
- Conectores a diferentes motores de procesamiento de datos, como Spark, MapReduce o Flink.
- Conectores a diferentes almacenes de datos, como Hbase o Presto.

Teniendo en cuenta sus componentes, Alluxio puede considerarse un optimizador de la gestión de recursos, por lo que puede convertirse en una pieza que se vaya encontrando en los ecosistemas anteriores a medida que madure.

El ecosistema de Apache Alluxio se estructura como se ilustra en la figura 17.

Figura 17. Ecosistema Apache Alluxio



Fuente: Apache Alluxio.

Alluxio es la empresa que da soporte a este ecosistema.

4.3.5. Ecosistema H_20

H_20 es un ecosistema focalizado en *machine learning*. Fundado por SriSatish Ambati en 2011, el objetivo es crear la mejor librería de analítica para *big data* usando cualquiera de los principales motores de procesamiento (en estos momentos, Hadoop y Spark).

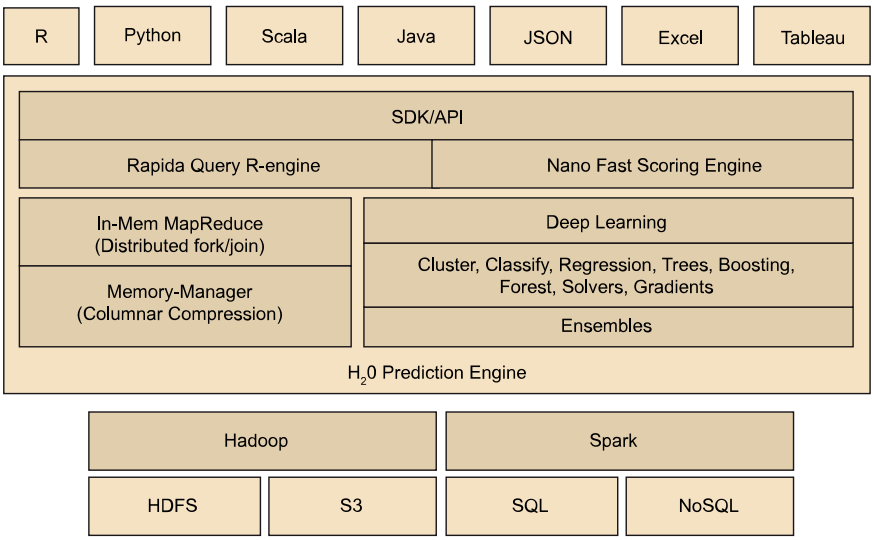
El ecosistema de H_20 se estructura en diferentes componentes:

- Conectores a diferentes sistemas de almacenamiento de datos, como Amazon S3, HDFS y cualquier base de datos relacional o NoSQL.
- Conectores a diferentes motores de procesamiento de datos, como Spark o Hadoop.
- Conectores a diferentes lenguajes de programación o de computación estadística, como R, Pyhton, Scala y Java.
- Conectores a herramientas de análisis, como Excel o Tableau.
- Motor de predicción, con foco en la inclusión de múltiples algoritmos.

Este ecosistema compite, por lo tanto, por la atención de los científicos del dato y cada vez tiene más competidores.

El ecosistema de H_20 se estructura como se ilustra en la figura 18.

Figura 18. Ecosistema H_20



Fuente: H_20.

4.3.6. Ecosistema Amazon

Google, Amazon y Microsoft ofrecen ecosistemas para *big data* en modalidad *cloud compluting*. Para ilustrar esta opción, vamos a presentar uno de ellos.

Amazon ofrece un ecosistema completo para *big data* a través de Amazon Web Services. Este ecosistema se caracteriza por ofrecerse en modalidad *cloud computing* y también en entornos híbridos. Cabe destacar que frecuentemente algunos de sus componentes se combinan con los anteriores ecosistemas.

Este ecosistema se estructura en tres grandes bloques (que no corresponden por completo a la taxonomía presentada en este material):

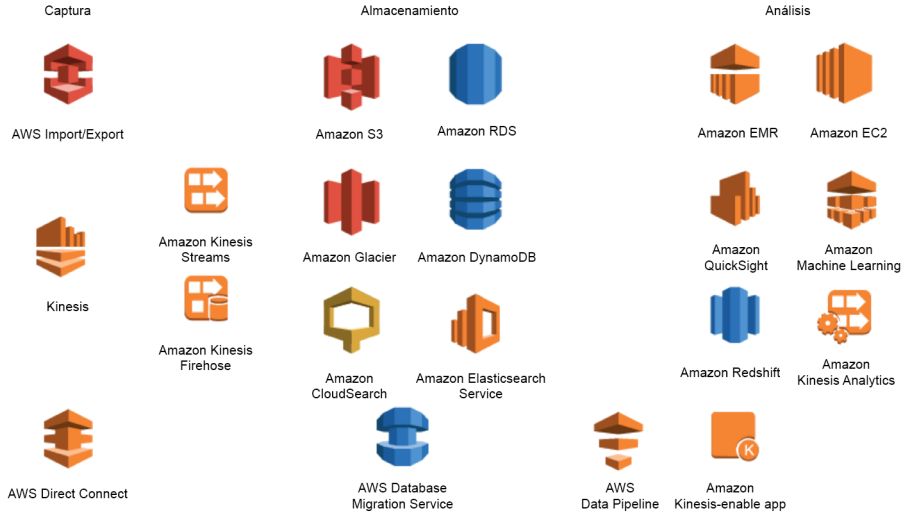
- **Captura del dato**: que habilita proceso ELT tanto en formato *batch* como en *streaming*.

- **Almacenamiento del dato**: que habilita el almacenamiento de datos en todo tipo de opciones (NoSQL, sistemas distribuidos, relacionales).

- **Análisis del dato**: que habilita el análisis de datos incluyendo inteligencia de negocio (Quicksight) y *machine learning*.

El ecosistema de Amazon se estructura como se ilustra en la figura 19.

Figura 19. Ecosistema Amazon big data



Fuente: Amazon.

4.4. Comparativa principales motores de procesamiento *big data*

A lo largo de este material se ha hablado de diferentes motores de procesamiento; la tabla 7 compara varios motores en cuanto a características técnicas como API, el paradigma de procesamiento, el tipo de optimización básico y el tipo de ejecución.

Tabla 7. Comparativa motores procesamiento

Característica	MapReduce	Tez	Spark	Flink
API	MapReduce aplicado a parejas *key-value* (k-v)	Lectura/Escritura de parejas k-v	Transformaciones en conjuntos de parejas k-v	Transformación iterativa de colecciones de datos
Paradigma	MapReduce	Directed-Acyclic-Graph (DAG)	Resilient Distributed Datasets (RDD)	Flujos de datos cíclicos
Optimización	Ninguna	Ninguna	Optimización para consultas SQL	Optimización para todo tipo de API
Ejecución	Ordenación *batch*	Ordenación y particionamiento *batch*	Procesamiento *batch* basado en memoria	Procesamiento *streaming* basado en memoria

DAG

Cuando hablamos de DAG hacemos referencia a un grafo en el que no hay ciclos directos.

4.4.1. Casos de uso ecosistemas

Hemos ilustrado tan solo algunos ecosistemas existentes en el mercado de *big data*. La existencia de tantos ecosistemas dificulta enormemente la selección de uno de ellos y, sobre todo, de uno de los múltiples fabricantes que ofrecen una plataforma así como de los integradores de servicio.

Existen ciertos puntos que cabe tener en cuenta para elegir un componente frente a otro. Destacamos algunas reflexiones:

- **Hadoop** se usa cuando tenemos datos en el rango de Terabytes o Petabytes y expectativas de crecimiento. Es decir, demasiada información para una única máquina; de manera que se usa HDFS para almacenar el dato y MapReduce para su procesamiento.

- Como ya sabemos, **Spark** emerge para mejorar las limitaciones de MapReduce. Spark ofrece una API más simple y más fácil de usar. En general, se apuesta por Spark exceptuando si ya existe un caso de uso con MapReduce y no existe la intención de migración o por el hecho de que Spark no escale correctamente. Hay escenarios en los que Spark falla con grandes volúmenes de datos al ser una tecnología más moderna, aunque cada vez hay menos errores. En esencia, Spark es una tecnología de *batch processing* que puede aproximarse a ciertos escenarios de *streaming processing*. Es posible que un criterio similar se aplique en los próximos años para **Apache Flink**, aunque en este caso hay menos empresas con conocimiento.

- **Apache Kafka**, creado en 2011 por LinkedIn, es un sistema de cola como RabbitMQ o Apache ActiveMQ, pero distribuido y permite trabajar con datos en movimiento. En el caso de que el dato exceda las capacidades de procesamiento de Kafka, el sistema los puede almacenar para su posterior tratamiento. Además, este componente puede usarse como multiplicador replicando el dato para diferentes aplicaciones. Mientras que Kafka permite capturar y almacenar datos en *streaming*, **Apache Storm** permite su procesamiento y aplicar lógica de negocio. Por ejemplo, revisar una transacción de una tarjeta de crédito y decidir si se acepta o se rechaza. Su principal competidor es Spark Streaming. Storm soporta *at-least-one semantics*, lo que significa que un mensaje puede procesarse más de una vez si la máquina falla. Por otro lado, Spark Streaming soporta *exactly-one semantics*, lo que significa que el mensaje se procesa una vez. Si se necesita procesar el dato de manera inmediata se elegirá Storm. Si se acepta una latencia mayor (varios segundos), se puede usar Spark Streaming (puesto que trabaja con lotes). El que será potencialmente el remplazo de Storm es Apache Flink, pero actualmente es una tecnología todavía con pocos casos en producción.

- **Hbase** tiene la misma funcionalidad que HDFS pero una gran diferencia: permite modificar los registros guardados. Por tanto, sus casos de uso están más vinculados a almacenar la información más reciente y su análisis. Frecuentemente compite con Cassandra. En general, Hbase es más rápido en lectura que en escritura comparado con Cassandra. En el caso de Hadoop, Hbase suele ser la opción preferente. Cassandra es la opción preferida cuando Hadoop no sea necesario.

- **Hive** permite traducir las consultas SQL en procedimientos MapReduce. Existen componentes similares, como Pig, Impala o SparkSQL. Cuando el dato sea estructurado, Hive será la opción, y cuando no lo sea, Pig. Impala es más rápido pero solo está disponible a través de la plataforma de Clou-

dera. SparkSQL es un componente de Spark, por lo que se suele usar en el caso de estar implementando Spark. En general, Hive es para procesamiento *batch* nocturno, mientras que Impala o SparkSQL es para análisis exploratorio.

5. Anexo

5.1. Fuentes abiertas de datos

Como se ha comentado, las fuentes externas de datos son un recurso relevante para los proyectos de *big data*. A continuación listamos algunas de las fuentes más interesantes.

- Enigma: http://enigma.io/
- Kaggle Datasets: https://www.kaggle.com/datasets
- Yahoo Datasets: https://webscope.sandbox.yahoo.com
- Million Songs Dataset: http://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset
- Open Football: https://openfootball.github.io
- Global Earthquake Archive: http://www.emidius.eu/GEH
- R Datasets: http://stat.ethz.ch/R-manual/R-patched/library/datasets/html/00Index.html
- Rstudio Babynames: https://github.com/hadley/babynames
- Rstudio Fueleconomy: https://github.com/hadley/fueleconomy
- Rstudio NASA Weather: https://github.com/hadley/nasaweather
- Rstudio NYC Flights: https://github.com/hadley/nycflights13
- DataMartker Time Series Data Library: https://datamarket.com/data/list/?q=provider:tsdl
- Datasets from the book: A Handbook of Small Data Sets: http://www.stat.ncsu.edu/research/sas/sicl/data
- M & M Data: http://www.math.uah.edu/stat/data/MM.html
- Marvel Universe Social Graph: http://exposedata.com/marvel
- UCI Network Data Repository: http://networkdata.ics.uci.edu/index.php
- UC Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml
- The Mondial Database: http://www.dbis.informatik.uni-goettingen.de/Mondial
- Europa Open Data: http://open-data.europa.eu/es/data
- USA Data Gov: http://www.data.gov
- Facebook Graph API: https://developers.facebook.com/docs/graph-api
- Quandl: https://www.quandl.com/help/getting-started
- Dbpedia: http://wiki.dbpedia.org
- Stanford Large Network Dataset Collection: http://snap.stanford.edu/data/index.html
- UK Data Service: http://www.ukdataservice.ac.uk
- Open Data Network: http://www.opendatanetwork.com

Resumen

En este módulo didáctico hemos presentado el concepto de *big data*, que fundamentalmente habilita a las organizaciones a trabajar con conjuntos de datos complejos.

Hemos presentado su definición, los tipos que existen, qué beneficios aporta, cuándo es necesario aplicar esta estrategia y las tecnologías que forman parte de ella. Y sobre todo, se ha dejado de manifiesto cuán diferente y por qué complementa a la factoría de información corporativa.

Además, se ha hecho hincapié en la necesidad de creación de una estrategia para *big data* y en el uso de modelos de madurez para comprender en qué estado se encuentra una organización y si está preparada para abordar este tipo de proyectos.

Finalmente, se han discutido casos de uso y ejemplos. Tal y como se ha mostrado en estos materiales, existen multitud de organizaciones que ya han desplegado este tipo de sistemas de información y han conseguido rendimientos de la explotación de conjuntos de datos complejos.

Glosario

ACID *m* Estándar *de facto* de las bases de datos relacionales. Es el acrónimo de *atomicity* (atomicidad), *consistency* (consistencia), *isolation* (aislamiento) y *durability* (durabilidad).

API *m* Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

BASE *m* Estándar para las tecnologías *big data*. Es el acrónimo de *basically available* (básicamente disponible), *soft state* (estado blando) y *eventual consistency* (consistencia eventual).

big data *m* Conjunto de estrategias, tecnologías y sistemas para el almacenamiento, procesamiento, análisis y visualización de datos complejos.

business intelligence *m* Conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización.

Byte *m* Unidad de medida de información digital.

CAP *m* Estándar *de facto* de los sistemas distribuidos. Es el acrónimo de *consistency* (consistencia), *availability* (disponibilidad) y *partition tolerance* (tolerancia a la partición).

CEP *m* Procesamiento de eventos en tiempo real que combinan múltiples fuentes y que se usa para inferir eventos o patrones que siguieren situaciones complicadas como oportunidades y/o amenazas.

clúster *m* Conjunto de ordenadores conectados en red que trabajan de manera conjunta. Cada ordenador del clúster es llamado nodo.

CRM *m* Acrónimo de *customer relationship management*, que hace referencia a la gestión de la relación con clientes.

crowdsourcing *m* Proceso de obtener servicios, ideas, contenido a través de la participación de una gran masa de personas.

cuadro de mando *m* Sistema que informa de la evolución de los parámetros fundamentales de negocio de una organización o de un área de esta.

DAG *m* Acrónimo de *directed acyclic graph*. Hace referencia a un grafo en el que no hay ciclos directos.

data integration *f* Conjunto de aplicaciones, productos, técnicas y tecnologías que permiten una visión única y consistente de nuestros datos de negocio. También denominada integración de datos.

data warehouse *m* Repositorio de datos que proporciona una visión global, común e integrada de los datos de la organización, independiente de cómo se vayan a utilizar posteriormente por los consumidores o usuarios, con las propiedades siguientes: estable, coherente, fiable y con información histórica.

ecuaciones diferenciales *f pl* Ecuación matemática que relaciona una función y sus derivadas.

edge analytics *f pl* Aplicaciones analíticas para IoT en las que ciertos algoritmos se ejecutan en los nodos de la red y no solo en el centro de datos.

ERP *m* Acrónimo de *enterprise resource planning*, que hace referencia a la gestión de los recursos de una organización.

escalabilidad *f* Habilidad de un sistema, red o proceso o bien para reaccionar y adaptarse sin perder calidad, o bien para manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

escalabilidad horizontal *f* Escalabilidad fundamentada en el incremento de nodos del sistema, proceso o red.

escalabilidad vertical *f* Escalabilidad fundamentada en añadir más recursos (memoria, disco duro y/o procesadores).

estructura de datos relacional *f* Tipo de base de datos que permite establecer interconexiones o relaciones entre los datos guardados en tablas.

grid *m* Conjunto de ordenadores conectados en red que trabajan de manera conjunta pero, a diferencia del clúster, los ordenadores son heterogéneos, realizan tareas independientes o no están en la misma localización.

latencia *f* Suma de retardos temporales en la captura, el almacenamiento, el procesamiento y el análisis del dato.

internet de las cosas *m* Interconexión digital de objetos cotidianos con internet. Nos referiremos a ella por su acrónimo en inglés IoT, *internet of things*.

metadatos *m pl* Datos estructurados y codificados que describen características de un objeto, dato o proceso de negocio.

NIST *m* Acrónimo de National Institute of Standards and Technology, institución americana que estudia, define y promueve estándares tecnológicos.

NPL *m* Acrónimo de *natural processing language*. Hace referencia a un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

NoSQL *m* Acrónimo de *not only SQL*. Hace referencia a bases de datos no relacionales.

HPC *f* Práctica de añadir capacidad de computación de forma que mejora el rendimiento de una estación de trabajo y es posible abordar problemas complejos en la ciencia, ingeniería y/o negocios.

OLAP *m* Método para organizar y consultar datos sobre una estructura multidimensional. Es el acrónimo de *online analytical processing* o proceso analítico en línea.

open data *m* Conjuntos de datos considerados que son un bien común y, que por ello, son gratuitos, accesibles y bien estructurados para su descarga y análisis.

OWL *m* Es el acrónimo de *Web Ontology Language*. Hace referencia a un estándar para el diseño de ontologías de modelos de datos.

PMML *m* Acrónimo de *Predictive Model Markup Language*. Hace referencia a un estándar para el intercambio de datos entre organizaciones.

RIF *m* Acrónimo de *Rule Interchange Format*. Hace referencia a un estándar para el intercambio de datos entre organizaciones.

SCV *f* Acrónimo de *speed*, *consistency* y *volume*. Es decir, a la velocidad de procesamiento, a la exactitud del dato y a la cantidad de datos procesados.

SLA *m* Acuerdo que estipula el nivel de servicio, el soporte, las posibles penalizaciones, el nivel de alta disponibilidad tanto de hardware como de software y el precio.

SQL *m* Acrónimo de *Structure Query Language*, hace referencia al lenguaje de consultas de bases de datos relacionales.

taxonomía *f* Clasificación u ordenación en grupos de cosas que tienen unas características comunes.

UIMA *f* Es el acrónimo de *Unstructured Information Management Architecture*. Hace referencia a un estándar que permite la interoperabilidad de analítica de datos en información no estructurada.

variabilidad *f* Característica de los flujos de datos que supone que pueden tener comportamientos erráticos o inconsistentes en ciertos periodos.

velocidad *f* Hace referencia tanto al procesamiento de datos como a su latencia.

variedad *f* Se refiere tanto a la cantidad de fuentes diferentes que combinar como a la heterogeneidad del dato.

veracidad *f* Incertidumbre en el dato producto de su baja calidad, la ambigüedad en su definición o simplificaciones en su modelización.

vinculación *f* Dificultad de relacionar diferentes y dispares fuentes de datos.

volumen *m* Tamaño del conjunto de los datos creado diariamente.

XBRL *m* Acrónimo de *eXtensible Business Reporting Language,* hace referencia a un estándar para informes financieros.

Bibliografía

Davenport, T. H. (2014). *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities*. Boston: Harvard Business Review Press.

Davenport, T. H.; Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning*. Nueva York: Harvard Business Press.

Davenport, T. H.; Kim, J. (2013). *Keeping Up with the Quants: Your Guide to Understanding and Using Analytics*. Boston: Harvard Business Review Press.

Erl, T.; Khattak, W.; Buhler. P. (2015). *Big Data Fundamentals: Concepts, Drivers & Techniques*. New Jersey: Prentice Hall.

Fisher, T. (2009). *The Data Asset: How Smart Companies Govern Their Data for Business Success*. New Jersey: Wiley.

Malcolm, F.; Roehrig, P.; Pring, B. (2014). *Code Halos: How the Digital Lives of People, Things, and Organizations Are Changing the Rules of Business*. New Jersey: Wiley.

Foreman, J. W. (2013). *Data Smart: Using Data Science to Transform Information into Insight*. New Jersey: Wiley.

Redman, T. C. (2008). *Data Driven: Profiting from Your Most Important Business Asset*. Boston: Harvard Business Review Press.

Schmarzo, B. (2016). *Big Data MBA: Driving Business Strategies with Data Science*. New Jersey: Wiley.

Schmarzo, B. (2013). *Big Data MBA: Understanding How Data Powers Big Business*. New Jersey: Wiley.