Big data

Àlex Caminals Sánchez de la Campa

PID_00197296

UOC
Universitat Oberta de Catalunya

Índice

Introducción

El aumento de la información disponible en cualquier organización es una realidad en nuestra sociedad. Poder extraer conocimiento de esta información que permita la toma de decisiones basadas en hechos con el objetivo de ser más eficientes y competitivos es ya una necesidad más que un complemento a la toma de decisiones.

La inteligencia de negocio (BI) tradicional es capaz de proporcionar soluciones tecnológicas para la toma de decisiones hasta cierto punto. Sin embargo, existen limitaciones cuando hay grandes volúmenes de datos, cuando el análisis de esos datos requiere una gran velocidad de proceso, y cuando los datos carecen de una estructura homogénea simple.

Big data agrupa un conjunto de tecnologías y técnicas que nos permiten disponer de capacidad de análisis en casos donde haya gran cantidad de datos, altamente heterogéneos y de distintas fuentes. *Big data* está aún en fase de evolución, pero augura un gran potencial, siendo objeto de inversiones muy significativas por parte de los grandes fabricantes de BI.

El objetivo de este módulo es ofrecer información sobre *big data* desde el punto de vista conceptual, sin entrar de lleno en la tecnología que hay detrás. A pesar de esto, se introducen ciertos aspectos tecnológicos que se consideran clave para poder entender cómo se comportan los sistemas de BI de *big data* y las diferencias que existen entre estos y los sistemas BI tradicionales.

Se incluyen también ejemplos y escenarios para facilitar la comprensión de los conceptos introducidos. Además, también se presenta información sobre cómo calcular métricas de evaluación del proyecto, como el *ROI* y el *Payback Period* aplicados a proyectos de *big data*. Finalmente, se incluyen casos de éxito que ofrecen una visión real del BI basado en *big data*.

1. Introducción y conceptos básicos

1.1. Definición

Big data es el término acuñado para referirse a un conjunto de técnicas y tecnologías que permiten el análisis de datos en un contexto concreto donde una solución de BI tradicional no puede obtener los resultados deseados en cuanto a la obtención y el tratamiento de datos.

Este contexto, conocido como de las **3 V** (**las tres uves**), se basa en las siguientes características:

- **Volumen**: Las fuentes de datos contienen un volumen de datos muy por encima de los volúmenes habituales dentro de un sistema BI tradicional. En un sistema tradicional es normal hablar de volúmenes diarios medidos en megabytes y hasta pocos gigabytes. Podemos considerar un volumen de pocos gigabytes un límite para el BI tradicional, debido a las técnicas de implementación de las soluciones y a la tecnología usada. Sin embargo, existen fuentes de datos que generan volúmenes que superan ampliamente los pocos gigabytes de información diaria. En este caso, nos encontraremos ante un escenario donde el BI tradicional no será capaz de procesar tanto volumen de datos[1]. Este escenario es un caso típico donde podemos aplicar *big data*.

- **Velocidad**: La generación de datos puede ser muy veloz en los sistemas origen. Por ejemplo, las imágenes de vídeo grabadas por varias videocámaras de vigilancia en un circuito cerrado de televisión o los mensajes subidos por millones de usuarios en una red social generan información de manera continua. Y esta información puede ser susceptible de ser analizada. En estos casos, una solución de BI tradicional no sería capaz de incorporar esos datos al *data warehouse* a un ritmo suficiente como para satisfacer la demanda de información. Por ejemplo, si nuestra solución de BI debe analizar las imágenes de vídeo de un aparcamiento para detectar posibles delitos (robos de vehículos o atracos por ejemplo) en tiempo real, el tiempo de proceso y análisis de las imágenes debe ser mínimo (debe tender a cero). Este escenario implica la necesidad de nuevas técnicas y tecnologías para poder analizar los datos más rápidamente. Es decir, obtenerlos y tratarlos para su análisis posterior. Este escenario es donde deberemos aplicar *big data*.

- **Variedad**: Los datos necesarios para el análisis decisional pueden ser de cualquier tipo y tener cualquier formato. Normalmente entendemos por datos cualquier información estructurada, es decir, que sigue una estruc-

[1] Esta afirmación no debe tomarse como una norma estricta. Cierto es que un hardware potente puede ser capaz de tratar volúmenes de datos de varios gigabytes durante la carga de datos. Sin embargo, en este módulo veremos cómo en algunas circunstancias un hardware potente puede no ser suficiente para analizar los datos de manera que se cumplan todos los requerimientos de los usuarios.

tura conocida y más o menos homogénea, como por ejemplo los datos de una persona, que contienen distintos campos para indicar sus propiedades: nombre, apellidos, fecha de nacimiento, edad, etcétera. Sin embargo, los datos también pueden venir organizados en estructuras complejas, o de forma semiestructurada o desestructurada. Ejemplos de estos datos serían, el texto libre, datos de sensores, de sonido, de fotografías, de imágenes de vídeo y los ficheros de *log* entre muchos otros. En el BI tradicional, estos tipos de datos suelen dejarse de lado al carecer de mecanismos para extraer de ellos información útil para las organizaciones. *Big data* nos promete facilitar la extracción y tratamiento de estos tipos de datos para su posterior análisis, abriendo una gran ventana para el análisis de datos de cualquier tipo.

1.2. Identificación de un escenario *big data*

Cuando nos hallamos frente a un escenario con grandes volúmenes de datos, con una velocidad de generación de datos muy alta o con la necesidad de tratar datos de todo tipo, ya sean estructurados simples, complejos, semiestructurados o desestructurados, estaremos ante un escenario típico para aplicar una solución de BI basada en *big data*.

Es importante destacar que el mero hecho de encontrarnos con tan solo uno de estos casos ya es suficiente para considerar el proyecto como susceptible de aplicar una solución BI de *big data*. No será necesario encontrar los tres casos simultáneamente para poder tener esta opción en consideración.

Llegados a este punto, podríamos plantearnos la cuestión: ¿Puede *big data* aplicarse en cualquier proyecto BI? El razonamiento para dicha cuestión es:

- Si *big data* puede trabajar con grandes volúmenes de datos, también puede trabajar con volúmenes más reducidos.

- Si *big data* puede obtener y tratar información de manera rápida y eficaz consumiendo datos generados de manera continua, también puede trabajar con velocidades inferiores.

- Si *big data* puede analizar cualquier tipo de datos, ya sean estructurados simples, complejos, semiestructurados o desestructurados, evidentemente puede trabajar con los datos habituales (estructurados simples) de un sistema BI tradicional.

La respuesta es que *big data* puede utilizarse en cualquier escenario donde queramos analizar información generada por uno o varios sistemas de información. Sin embargo, utilizar *big data* para dar solución a unos requerimientos

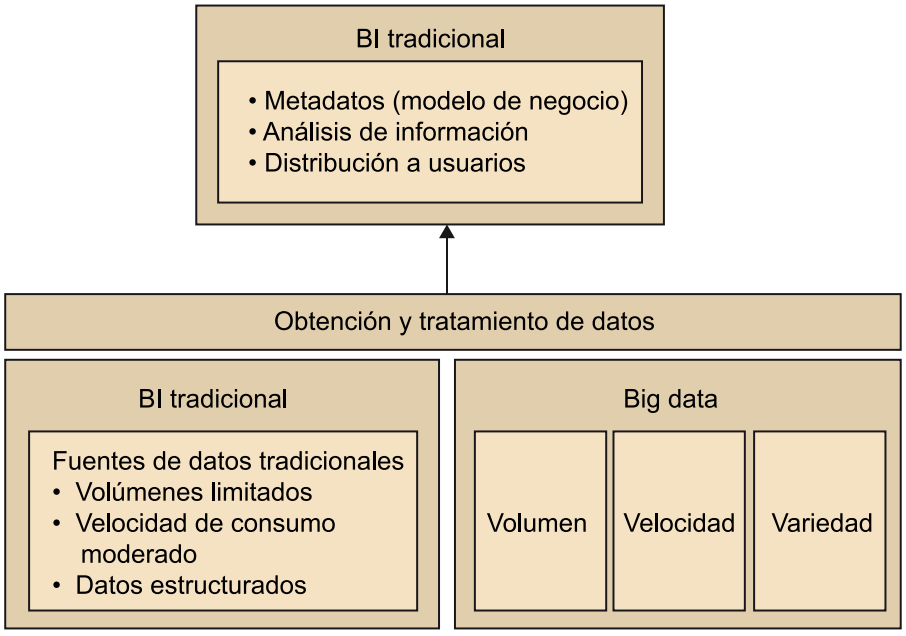
de análisis de datos cuando estos requerimientos pueden ser satisfechos por una solución BI tradicional supone un incremento de complejidad y de coste con respecto a la solución BI tradicional.

El hecho de utilizar *big data* no quiere decir que no vayamos a necesitar un sistema de BI tradicional. De hecho, son complementarios. Por su parte, *big data* se encargará de la obtención, tratamiento y análisis de información concreta (basada en las 3 V), mientras que el BI tradicional se encargará de lo mismo para información con la cual pueda ofrecer los resultados deseados. Además, del BI tradicional también obtendremos las capas superiores de nuestra solución de BI, componentes indispensables en la arquitectura de cualquier sistema BI.

Una solución *big data* debe utilizarse como complemento a un sistema BI tradicional, en los casos en que esta no pueda proporcionar una solución a los requerimientos analíticos de los usuarios.

La convivencia entre el BI tradicional y *big data* se ilustra en el siguiente esquema:

Complementariedad de *big data* con un sistema BI tradicional



Si lo que buscamos son unos límites que nos permitan identificar cuándo debemos apostar por una solución de *big data*, no vamos a encontrar una respuesta. Lo importante es evaluar el potencial de la solución BI tradicional a usar con referencia a las necesidades de análisis de la solución BI que necesitamos. Es decir, si con un conjunto de herramientas tradicionales podemos capturar y tratar los datos de que dispondremos. Esto debe hacerse en base a las tres dimensiones que hemos visto anteriormente: volumen, velocidad y

variedad de los datos. En el caso de que las herramientas y técnicas tradicionales no nos permitan obtener los resultados deseados, deberemos optar por una solución *big data*.

Por tanto, no existe un volumen de datos concreto, ni una velocidad de necesidad de proceso de datos, ni una definición de variedad de datos que marque la frontera entre no usar o usar *big data*. Todo dependerá de la capacidad de la solución BI tradicional para proporcionar los resultados deseados.

1.3. Diferencias entre *big data* y un sistema BI tradicional

Las diferencias a grandes rasgos entre *big data* y un sistema BI tradicional son:

1) Naturaleza de los datos: *big data* se utiliza para obtener y tratar:

- grandes volúmenes de datos,
- datos generados a una velocidad muy alta que requieren su análisis en un tiempo mínimo,
- datos sin una estructura uniforme.

2) Almacenamiento y uso de los datos: En un sistema BI tradicional, los datos son almacenados en su mínima granularidad en el *data warehouse*. Esto permite el análisis de los datos en las mismas consultas. Por ejemplo, podemos analizar las ventas de los últimos tres meses para una línea de producto concreta. Esto se realiza a partir de la agregación de los datos de las ventas almacenados en su mínima granularidad en el *data warehouse*. Si bien es cierto que, para aumentar el rendimiento de las consultas y reducir el tiempo de respuesta de los análisis, podemos crear agregaciones en el *data warehouse*, estos se basan en los datos presentes en el mismo *data warehouse*. Esta disponibilidad de los datos nos permite hacer *drill-down* en los análisis.

Granularidad

La granularidad de los datos indica el nivel de detalle de estos. La similitud con los granos de arena es perfecta para ilustrar el concepto. Mínima granularidad significa que vemos los datos como granos de arena individuales. O sea, que tenemos el máximo nivel de detalle de estos. Sin embargo, cuando empezamos a juntar granos de arena para hacer una montaña de granos de arena, la granularidad aumenta. O lo que es lo mismo, el nivel de detalle disminuye.

Utilizando un escenario de negocio, una llamada telefónica es indivisible, es la mínima unidad, lo que implica mínima granularidad. Ello nos permitiría obtener el tiempo de conversación de cada una de las llamadas telefónicas de manera individual. Por otra parte, analizar el tiempo total de conversación entre todas las llamadas telefónicas realizadas durante un día implica una mayor granularidad, ya que el nivel de detalle es inferior.

En el caso de *big data*, bien sea por volumen, por la velocidad en que son generados o porque no tienen estructura, los datos, en su mínima granularidad, no suelen formar parte del *data warehouse*. La información que se incluye en el *data warehouse* es información derivada de los datos obtenidos y tratados por *big data*, información previamente filtrada y agregada en la mayoría de los casos. Esta información formará parte del *data warehouse* y podrá utilizarse para el análisis decisional dentro del área de negocio a la cual pertenezcan los datos.

Por ejemplo, una organización puede obtener información de las distintas redes sociales para averiguar el sentimiento asociado a un nuevo producto recién lanzado al mercado. En lugar de guardar en el *data warehouse* todos los datos en su mínima granularidad (comentarios en Facebook, tweets, fotos, vídeos

de YouTube, etc.), los datos se tratarán mediante técnicas de *big data* (como por ejemplo técnicas de opinión *mining*) y se derivará un conjunto de datos iniciales. Teniendo en cuenta la actividad de las redes sociales, podemos imaginar que el sentimiento u opinión respecto al nuevo producto puede medirse a intervalos de tiempo cortos. Ello nos permitirá analizar, por ejemplo, las correlaciones entre el sentimiento de la comunidad y las ventas, y cómo afectan en ellas los cambios en la web corporativa, notas de prensa, y noticias difundidas en los medios.

3) Tecnología: Si pudiéramos utilizar la misma tecnología para poder obtener y tratar datos en un contexto 3 V de *big data*, no sería necesario haber desarrollado ninguna de las técnicas ni de las tecnologías que hoy día conforman *big data*. Posiblemente necesitaríamos máquinas más potentes, pero no un cambio tecnológico. La realidad es que la tecnología utilizada en el BI tradicional no puede soportar el contexto de 3 V de *big data*. Es por eso por lo que se han incorporado nuevas tecnologías.

Big data es una área aún en evolución y nuevas soluciones son creadas o mejoradas y añadidas continuamente a paquetes de software para *big data*. Los grandes fabricantes de software compran empresas con componentes muy especializados que permiten ampliar las capacidades de sus paquetes de software de *big data*.

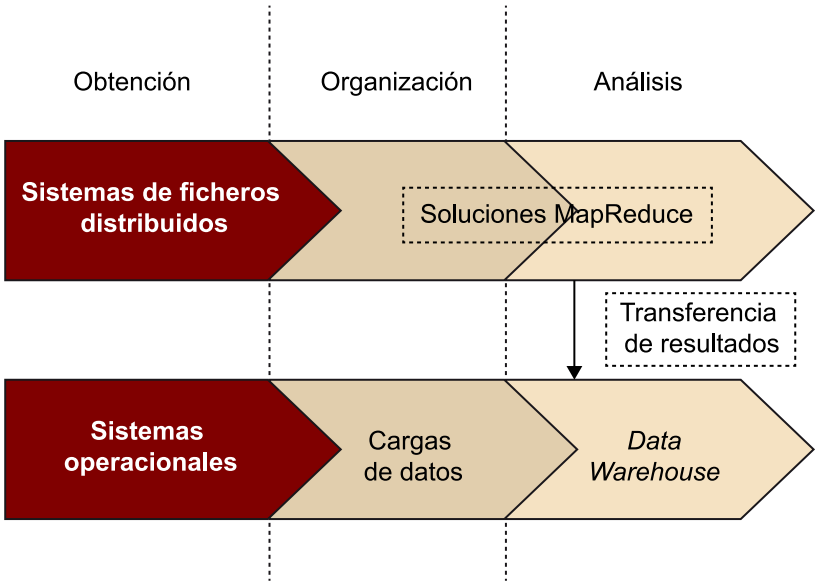
La tecnología usada en *big data* consiste básicamente en un sistema de ficheros distribuido que almacena los distintos tipos de datos utilizados en la solución BI (imágenes, fotografías, vídeo, audio, ficheros de texto desestructurados, etc.), y un conjunto de herramientas conocidas genéricamente como **Soluciones MapReduce**, que permiten la organización y análisis de los datos. De manera opcional los resultados obtenidos en el análisis de información en *big data* pueden ser transferidos al *data warehouse*. Esto permite la integración de los resultados del análisis de *big data* con los disponibles en el *data warehouse*, lo cual conlleva un mayor potencial de análisis.

La tecnología de un sistema de BI con *big data* puede resumirse en el siguiente diagrama.

Drill-down

Drill-down es el término utilizado para designar la navegación a niveles de información inferior dentro de un informe. Por ejemplo, si tenemos un informe con el número de accidentes de tráfico por comunidad autónoma, podemos hacer una navegación al siguiente nivel de detalle para ver ese mismo informe, no por comunidad autónoma sino por provincia. En este caso hablaríamos de un *drill-down* del primer informe a nivel de provincia.

Tecnología en un entorno BI con *big data*



Ved también

En el apartado "Tecnología" se detalla el contenido de cada uno de los componentes mostrados en el diagrama "Tecnología en un entorno BI con *big data*".

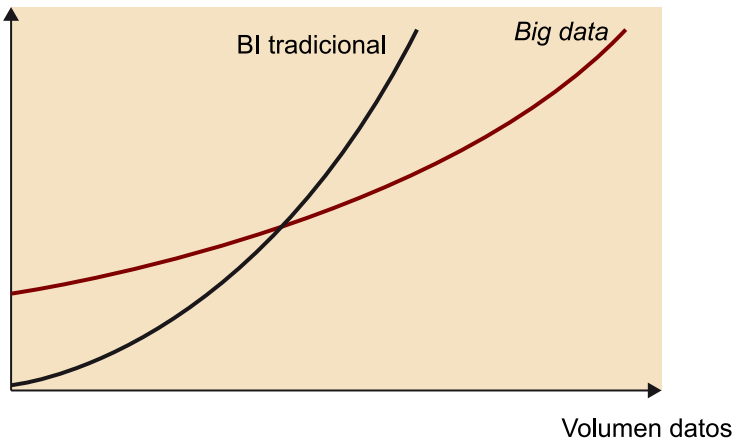
2. Incorporación de *big data* a un sistema BI tradicional

Un sistema BI no es estático. Los datos están vivos y los requerimientos cambian. Estos cambios a los que debe hacer frente toda solución BI se pueden resumir en los siguientes escenarios:

1) Aumento del volumen de los datos. Este es el caso obvio e intrínseco a todas las soluciones BI. Información nueva es generada en las fuentes de datos continuamente. Y esta información debe ser incluida en el *data warehouse* para su análisis. A medida que los datos crecen, el sistema BI debe hacer frente al problema de almacenar y procesar dichos datos para su posterior análisis de manera eficiente por los usuarios. Llegados a un punto en que un sistema BI tradicional no pueda ofrecer una solución analítica que albergue todos los datos, la solución debería ser modificada o hasta rediseñada para poder satisfacer los requerimientos analíticos de los usuarios. Sin embargo, en algún momento podemos llegar a un punto en que tanto la tecnología como las técnicas utilizadas no permitan ofrecer un resultado satisfactorio. En ese caso, nos plantearemos una evolución de la solución BI para incorporar *big data* en esa área donde el sistema BI tradicional no nos permita tratar los datos eficientemente.

El siguiente diagrama nos muestra cómo, a partir de cierto punto, una solución con *big data* resulta más efectiva que una solución de BI tradicional.

Aumento del tiempo de ejecución de las cargas con el volumen de datos

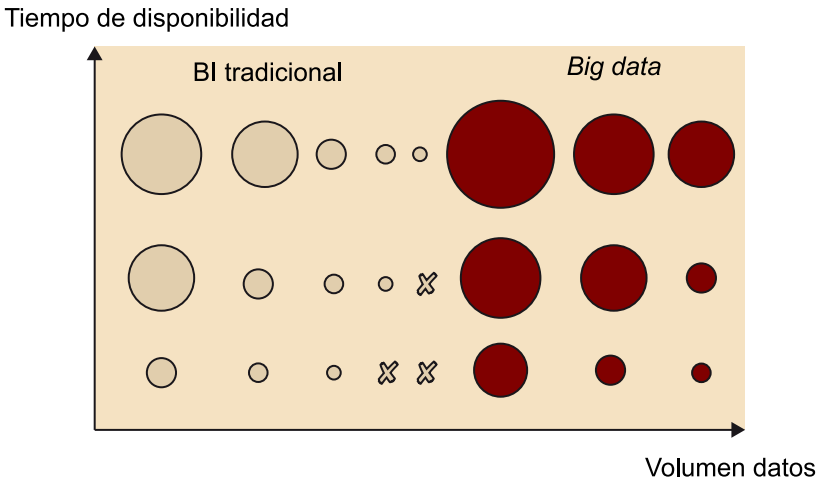


2) Aumento en la velocidad de disponibilidad de los datos. En el mundo competitivo en el que vivimos, tener la información tan pronto como sea posible puede ser la diferencia entre el éxito y el fracaso. Por eso, los usuarios pueden cambiar sus requerimientos iniciales para pedir la disponibilidad de los datos en un tiempo mucho menor que el inicialmente pactado (tradicionalmente un día de retraso en los datos, aunque dependerá de los procesos

a los que pertenezcan los datos). Si bien podemos modificar la frecuencia de las cargas de datos, esta reducción del tiempo de proceso tiene un límite. Entonces, quizá en algún momento deberemos plantearnos un rediseño de las cargas de datos, como por ejemplo una evolución a BI en tiempo real. Otra solución para disponer de los datos dentro del tiempo requerido por los usuarios es implementar *big data*. Sin embargo, mientras una solución BI en tiempo real puede funcionar de manera muy eficiente con cantidades de información no muy elevadas, cuando los volúmenes de datos son importantes, *big data* da un paso al frente, posicionándose como la mejor alternativa.

El siguiente diagrama nos compara la efectividad de un sistema BI tradicional y *big data* en función de los volúmenes de datos y la velocidad de disponibilidad de los datos. El volumen de la esfera indica el nivel de efectividad de las cargas de datos a la hora de tratar ciertos volúmenes de datos en un tiempo concreto. A mayor esfera, más efectiva es la técnica. Podemos observar cómo el tiempo de disponibilidad es directamente proporcional a la efectividad de la técnica mientras que el volumen de datos es inversamente proporcional a esta. La gran diferencia radica en que la efectividad es mayor cuando utilizamos *big data*.

Efectividad en función de volúmenes de datos y disponibilidad



3) Diversificación de los datos. Nuevas fuentes de datos pueden ser incorporadas a la solución BI inicial. Si los datos son estructurados o cuentan con un nivel de estructura que nos permite fácilmente adaptarlos a las estructuras del *data warehouse*, podremos continuar con nuestra solución BI simplemente añadiendo procesos a las cargas existentes para tratar los nuevos datos. Sin embargo, cuando los nuevos datos sean complejos, desestructurados, o no permitan su importación en estructuras fijas de una manera simple, deberemos usar otros métodos para incorporar esos datos al *data warehouse* para su análisis. Por ejemplo, si disponemos de documentos de texto, imágenes fotográficas, de vídeo, archivos de sonido u otros formatos que solamente puedan ser incorporados a una base de datos mediante un campo binario que almacene su contenido (o mediante un enlace a un fichero), una base de datos como las típicamente usadas en un *data warehouse* no podrá ofrecer capacidades de

análisis para esta información. En este caso, *big data* nos permite extraer información de estos nuevos tipos de datos, con los cuales podremos proporcionar una herramienta de análisis para los usuarios.

En el siguiente diagrama podemos observar algunos ejemplos de los diferentes tipos de datos que pueden ser procesados por un sistema BI tradicional y por *big data*. Como se puede observar, *big data* permite procesar todos los tipos de datos tradicionales además de una serie de tipos de datos adicionales.

Variedad de tipos de datos

Es evidente que hay muchos más tipos de datos que los representados en el diagrama. El objetivo del diagrama sin embargo no es listar todos ellos, sino mostrar las capacidades avanzadas de *big data* respecto a un sistema BI tradicional.

Cuando un sistema BI se encuentra con alguno de los anteriores escenarios, deberemos considerar seriamente la evolución de la parte de la solución BI existente afectada por dicho escenario a *big data*.

Ejemplo: Obtención de datos de sensores

Un sistema BI tradicional obtiene datos de un sensor de movimiento situado en un museo. Este dispositivo inserta registros en un fichero de texto cada 30". Estos registros son procesados por una carga de datos con una frecuencia de una ejecución cada minuto. El análisis de los datos se realiza para detectar si ha habido detección de movimiento en un dispositivo concreto de los más de mil existentes en todo el museo. Ello puede significar la existencia de un intruso en el museo (la decisión depende de la información de los turnos de vigilantes y sus recorridos estrictamente estipulados, también presentes en el *data warehouse*).

En un momento dado, el museo decide comprar nuevos sensores de movimiento y dispositivos de localización para los vigilantes para permitir unos recorridos más flexibles. Los nuevos sensores generan un registro cada milisegundo. Los localizadores de los vigilantes también. Esta frecuencia de muestras permite un análisis de los datos más detallado, con lo cual se decide analizar los datos cada dos segundos para detectar la posible existencia de intrusos.

El aumento del volumen y de la velocidad de análisis de los datos hace que el sistema BI existente en el museo no tenga la capacidad de procesar los datos generados por los nuevos dispositivos. En este caso, se podría optar por una solución *big data* para la obtención y tratamiento de los datos de los sensores de movimiento y de posicionamiento.

Sin embargo, el resto de datos incorporados al *data warehouse* en las cargas de datos tradicionales no se verían afectados por este cambio, ya que la solución existente es capaz de satisfacer los requerimientos iniciales de los usuarios y el sistema de *big data* puede integrarse al actual sin tener que sustituirlo completamente.

En resumen, un sistema BI tradicional que no pueda procesar datos para su análisis debido a la incapacidad de procesar grandes volúmenes de datos en un tiempo concreto, o porque carece de las herramientas para procesar ciertos tipos de datos, es un firme candidato para incorporar técnicas y tecnologías de *big data* para solucionar estas situaciones. En esos casos, *big data* se debe aplicar solo donde sea necesario y donde exista un beneficio para el usuario. En ningún caso se deberá hacer una migración total de la solución existente a *big data*, y menos aún eliminar la herramienta de BI tradicional, ya que esta es necesaria para la generación y presentación de los resultados de los análisis a los usuarios.

3. Tecnología

Big data combina una serie de técnicas y tecnologías que permiten obtener y procesar datos para su análisis más allá de las capacidades de un sistema BI tradicional.

Debido a las limitaciones existentes en los sistemas BI tradicionales para procesar según qué tipo o cantidad de datos de manera eficiente, es necesario un cambio de tecnología. La tecnología usada por *big data* sigue un paralelismo con la tecnología usada en proyectos BI tradicionales, ya que las funciones que busca son análogas. Estas funciones son las siguientes:

- **Obtener y almacenar los datos**: La información generada por las fuentes de datos debe ser obtenida y almacenada para su posterior análisis.

- **Interpretar y extraer información**: Una vez los datos han sido almacenados, los usuarios deben ser capaces de ejecutar consultas y obtener respuestas de esos datos. Para ello, es necesario convertir esos datos iniciales en información útil para los usuarios.

3.1. Obtención y almacenamiento de datos

A continuación se plantean las limitaciones de la tecnología tradicional en la función de obtener y almacenar los datos para dar paso a la presentación de la tecnología usada en *big data*.

3.1.1. Inconvenientes de un sistema BI tradicional

La tecnología usada históricamente para la obtención y el almacenamiento de los datos es el **sistema gestor de bases de datos relacional (SGBDR)**. Los SGBDR permiten la creación de modelos transaccionales y dimensionales dependiendo de la utilidad que le queramos dar a la base de datos. En ambos casos, existen posibilidades de ajuste de la base de datos para su optimización y eficiencia.

Uno de los principios básicos de un SGBDR es la integridad de los datos. Esto supone una sobrecarga en el momento de almacenar los datos. Por ejemplo, si queremos almacenar los datos de una venta, es lógico que queramos almacenar información sobre el cliente, el producto y el vendedor, entre otras dimensiones. Cada una de estas dimensiones, además de la propia venta, se almacenará en un contenedor diferente llamado tabla. Las tablas nos permiten almacenar y organizar los datos de la misma naturaleza (clientes, productos, vendedores, transacciones de venta, etc.), asegurándonos de que cada una

de estas entidades tiene las mismas características (llamadas columnas). Además, un SGBDR nos permite definir integridad referencial entre tablas. En este escenario, por ejemplo, eso significaría que una transacción de venta debe contener un producto, un cliente y un vendedor válidos que estén presentes en la base de datos. Aparte de la estructuración de la información en tablas, los sistemas relacionales acostumbran a gestionar las operaciones de inserción y modificación que reciben mediante transacciones. Simplificando, podemos decir que estas transacciones se encargan de garantizar que la información de la base de datos se modifica/añade de forma consistente, por ejemplo, provocando que si al dar de alta una venta el sistema no es capaz de añadir información sobre una de sus dimensiones (por ejemplo el cliente o sus datos de pago), entonces la transacción entera (toda la información relacionada con la venta) es rechazada.

Las ventajas de un SGBDR son evidentes, pero ello acarrea una sobrecarga en el tratamiento de los datos. Esto significa, para grandes volúmenes de datos que deben ser tratados en un espacio de tiempo muy corto, un inconveniente demasiado grande. En este caso, el resultado es que el SGBDR no es capaz de procesar y almacenar los datos a la velocidad necesaria.

Otro inconveniente de los SGBDR es la dificultad de almacenar y tratar tipos de datos complejos, como, por ejemplo, ficheros de audio, de vídeo y fotografías.

Ficheros de audio

Si bien es cierto que los ficheros binarios (.wav, .mp3, .wma, etc.) pueden almacenarse en una columna de una tabla de un SGBDR, esos datos carecen de estructura o sentido dentro del SGBDR. Es decir, tan solo son una cadena binaria y por tanto no permiten su análisis. Esto en el caso de un fichero .mp3 significaría la imposibilidad de obtener la información incluida en el contenido del mismo fichero, como pueden ser el título, el autor y el género musical, entre otros.

3.1.2. Hadoop

Big data no es un producto tecnológico sino un conjunto de técnicas y tecnologías destinadas a conseguir unos resultados hasta ahora no obtenidos con las técnicas y el software de BI tradicional.

Muchos han sido los que han trazado una ruta para proporcionar una solución técnica para la implementación de *big data*, pero no existe ninguna plataforma ni lenguaje estándar, como podría ser el estándar SQL para acceso a bases de datos relacionales.

Dentro de todos los proyectos existentes, hay uno que destaca por encima de todos ellos por su gran popularidad, su nombre es *Hadoop*.

Hadoop es un proyecto *open source* desarrollado por la Apache Software Foundation. El objetivo inicial del proyecto era crear un entorno de computación sobre un entorno distribuido. De esta manera, se podía utilizar la potencia de todos los nodos del sistema distribuido[2] para la computación de datos a gran escala.

(2)El conjunto de nodos que constituyen el sistema distribuido de Hadoop se denomina clúster.

Los cinco componentes básicos de Hadoop a día de hoy[3] son:

(3)Tal y como se ha dicho, al tratarse de un proyecto en marcha, Hadoop se halla en constante evolución. Así pues, es posible que nuevos módulos y componentes se hayan añadido a Hadoop entre el tiempo de escritura de este material y su lectura.

1) Hadoop distributed file system (HDFS): está formado por el conjunto de discos de los diferentes nodos del clúster de Hadoop.

Los datos en Hadoop son desmenuzados en lo que se llaman bloques. Estos bloques de datos son distribuidos entre los diferentes nodos que conforman el clúster. Esto permite que grandes volúmenes de datos puedan ser tratados en paralelo para así consolidar los resultados posteriormente.

Una de las ventajas que ofrece HDFS reside en su tolerancia a fallos de disco. Los diferentes bloques de datos son almacenados en diferentes nodos, lo que permite tener múltiples copias de los datos.

Otra gran ventaja es la de la cercanía de los datos. Al disponer cada nodo de datos almacenados en sus propios discos locales, el acceso a ellos es rápido y exclusivo, con lo que el rendimiento aumenta.

Arquitectura típica de Hadoop

La arquitectura típica de Hadoop se basa en un clúster formado por una gran cantidad de nodos con discos locales. Esto puede dar lugar a clústeres formados por pequeños servidores y computadoras personales, lejos de las grandes arquitecturas. De hecho, se recomienda no implementar Hadoop en sistemas de almacenaje avanzados/profesionales, ya que la sobrecarga en las comunicaciones podría llegar a causar cuellos de botella en estos sistemas.

2) Hadoop MapReduce model: MapReduce es una técnica de programación consistente en la transformación de datos (*map*) a una simplificación (*reduce*) de estos. Hadoop basa su capacidad de extracción de información de los datos originales en algoritmos MapReduce que se ejecutan en los diferentes nodos del clúster.

La técnica de programación responde al principio de "divide y vencerás", donde un problema es descompuesto en varios subproblemas. Una vez hemos solucionado los distintos subproblemas, es relativamente fácil solucionar el problema inicial a partir de la combinación de las soluciones de los subproblemas.

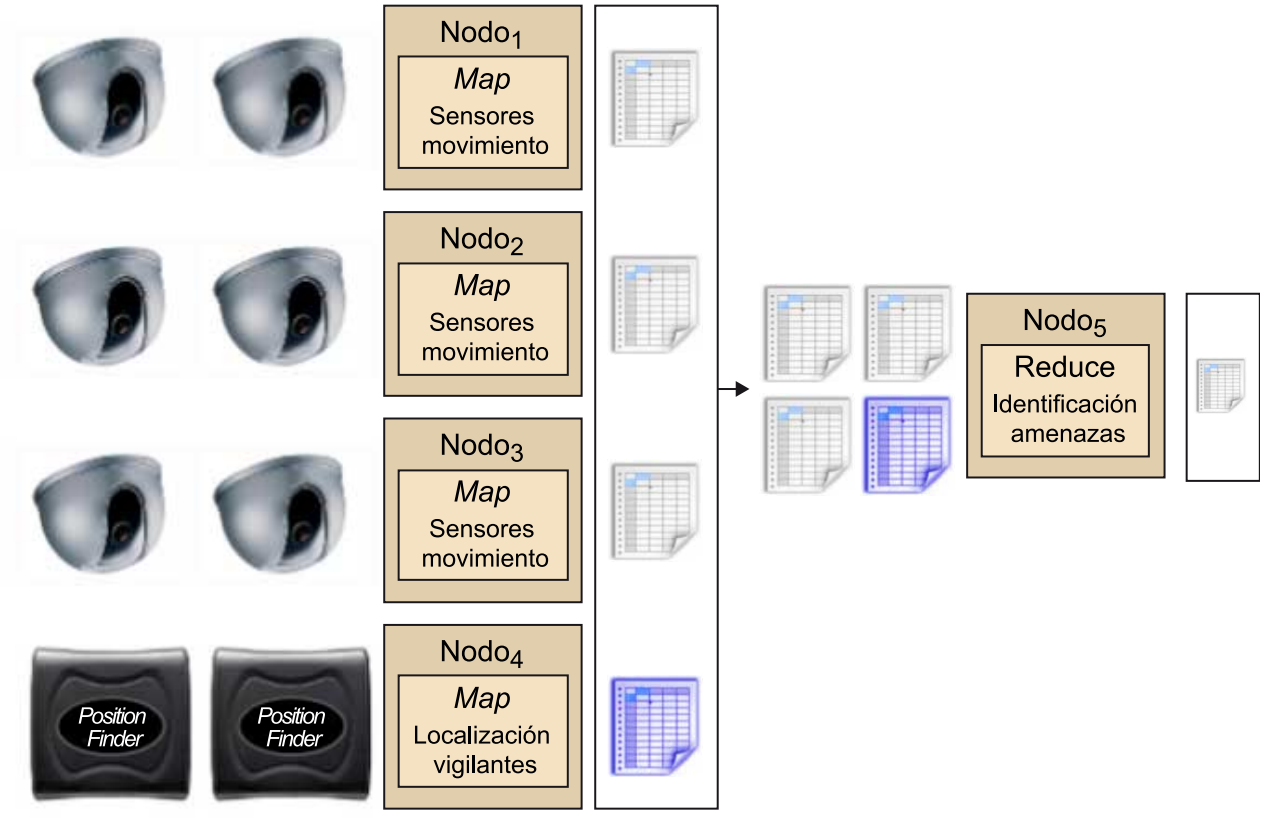
Ejemplo

Podemos hacer una analogía entre Hadoop y una cadena de montaje. En la construcción de un producto, por ejemplo un coche, también se aplica la técnica MapReduce. Construir un coche es un problema muy grande. Para simplificarlo, su construcción se divide en varios subproblemas (construcción de piezas, compra de componentes a terceros, etc.). Una vez ya hemos obtenido los componentes (o sea, hemos solucionado los subproblemas), estos deben ser simplemente ensamblados para obtener el resultado esperado inicialmente: la construcción de un coche.

Todos los programas MapReduce estan escritos en Java.

El siguiente diagrama muestra cómo grandes volúmenes de datos son tratados mediante la técnica MapReduce de Hadoop para obtener un reducido volumen de información útil para el usuario final. En concreto, los tres primeros nodos del clúster de Hadoop realizan el *map* de los datos de los cientos de sensores esparcidos por un museo. Por su parte, el nodo 4 se encarga del *map* de los datos de posicionamiento de los vigilantes. La información obtenida por los distintos nodos del clúster dentro de la fase de *map* son enviados a otro nodo que realiza el *reduce* de estos datos. Este nodo es el que identificará las posibles amenazas para la seguridad del museo, analizando en conjunto la información obtenida en la fase de *map*.

Ejemplo de Hadoop MapReduce



3) Hadoop YARN: YARN proporciona un marco para la gestión de recursos del clúster de Hadoop y para la planificación y control de los trabajos de ejecución.

4) Hadoop Common: Hadoop Common es el conjunto de librerías que soporta la ejecución de los diferentes componentes de Hadoop.

El proyecto *open source* de Hadoop sigue incorporando extensiones y nuevas funcionalidades. Varios ejemplos de estas son:

- Avro™: Sistema de serialización de datos.
- Cassandra™: Base de datos NoSQL escalable con copias redundantes que elimina los riesgos de pérdida de datos en el caso de problemas en la base de datos.
- Chukwa™: Sistema de obtención de datos para sistemas distribuidos.
- HBase™: Base de datos distribuida y escalable. Soporta almacenamiento de datos estructurados.
- Hive™: *Data warehouse* que permite la agregación de datos y consultas *ad hoc*.
- Mahout™: *Data mining*.
- Pig™: Lenguaje de alto nivel y entorno de ejecución para trabajar con flujos de datos.
- ZooKeeper™: Sistema de coordinación de servicios para plataformas distribuidas.

Debido a que es un proyecto activo y en pleno desarrollo y crecimiento, es prácticamente imposible decir qué componentes forman Hadoop de manera acotada.

Componentes Hadoop

Para una versión actualizada de los componentes incluidos en las diferentes versiones, lo ideal es visitar la página web del proyecto en Internet (hadoop.apache.org/).

3.1.3. Bases de datos NoSQL

Las bases de datos NoSQL (Not Only SQL, "no solo SQL") son la solución al problema del almacenamiento de grandes volúmenes de datos en un espacio de tiempo reducido.

Su diseño les permite acceder a todos los datos entrados a partir de una clave única, que se asigna para cada uno de ellos. Si comparamos esta técnica de almacenamiento con el de los SGBDR, veremos que en el caso de una base de datos NoSQL no existen procesos que velen por la integridad. Es decir, los datos no son analizados antes de su almacenamiento para, por ejemplo, comprobar que un campo numérico contiene caracteres válidos. Esto le permite procesar rápidamente los datos a medida que van llegando sin un coste adicional. El hecho de trabajar con pares clave-valor también permite el almacenamiento de cualquier tipo de datos.

Como en cualquier base de datos, el objetivo no es tan solo almacenar los datos sino poder explotarlos. Es decir, consultar los datos y extraer información útil para los usuarios del sistema. En este caso, podemos hacer una analogía

entre las bases de datos OLTP y las bases de datos NoSQL, ya que ambas están orientadas al almacenamiento eficiente de una gran cantidad de datos, a la vez que proporcionan unas capacidades de consulta de esos datos limitada.

Sistema OLTP

En el caso de un sistema OLTP, al estar basado en un SGBDR, no se pretende indicar que las capacidades de acceso a los datos con SQL sean limitadas, sino que, debido al diseño OLTP, el análisis de grandes volúmenes de datos no es eficiente y, por tanto, limita la capacidad práctica de análisis de los usuarios.

En el caso de una base de datos NoSQL, el análisis de los datos (de cualquier tipo de datos), requiere un esfuerzo de programación para poder interpretar los datos almacenados y extraer la información analítica deseada.

Llegados a este punto, es donde las limitaciones de las bases de datos NoSQL requieren las capacidades de un SGBDR. Si queremos proporcionar a los usuarios una capacidad de análisis superior a la de una base de datos NoSQL además de una infraestructura robusta, fácilmente manejable y segura, deberemos combinar ambas bases de datos: NoSQL para el almacenamiento de los datos y la obtención de información basada en consultas simples, y un SGBDR para las consultas complejas, basadas en los resultados extraídos de la base de datos NoSQL.

3.1.4. Solución tecnológica para una solución BI basada en *big data*

Basándose en Hadoop, varios fabricantes han creado paquetes de software que se adaptan al marco de trabajo de este. Lo mismo ocurre con la mayoría de proyectos *open source*. Estos paquetes de software incorporan componentes que complementan el marco tecnológico de Hadoop.

La combinación Hadoop + base de datos NoSQL + SGBDR + herramienta BI es la solución básica para un proyecto de BI basado en *big data*. Es decir, podemos considerar que en general una solución de BI basada en *big data* consta de:

- Un sistema de ficheros de datos distribuido y/o base de datos NoSQL.
- La implementación de Hadoop MapReduce.
- Un sistema gestor de bases de datos relacionales (SGBDR).
- Una herramienta BI tradicional.

Algunos ejemplos de paquetes de software *big data* de desarrolladores de software son:

1) Oracle - Oracle big data Appliance

- Hardware: Diseñado especialmente para la ejecución del software Oracle para *big data*. Consta de 18 servidores Sun, con una capacidad de disco total de 648 TB, 216 cores y 864 GB de memoria RAM en total.

- Cloudera Manager + Cloudera Distribution including Apache Hadoop (CDH): Cloudera es el software de Oracle que implementa Hadoop. Incluye extensiones propias de Oracle.

- Paquete estadístico R. *open source*, de Oracle.

- Oracle NoSQL Database Community Edition.

- Oracle Enterprise Linux + Oracle Java VM. Distribución del sistema operativo Linux de Oracle y máquina virtual de Java.

2) IBM - InfoSphere BigInsights

- Apache Hadoop: Incluye los proyectos *open source* Pig, Jaql, Hive, Hbase, Flume, Lucene, Avro, ZooKeeper y Oozie.

- Productos propios de IBM.

- Tratamiento y consulta de textos.

- Exploración de datos en una interfaz de tipo hoja de cálculo.

- Instalación y administración integradas.

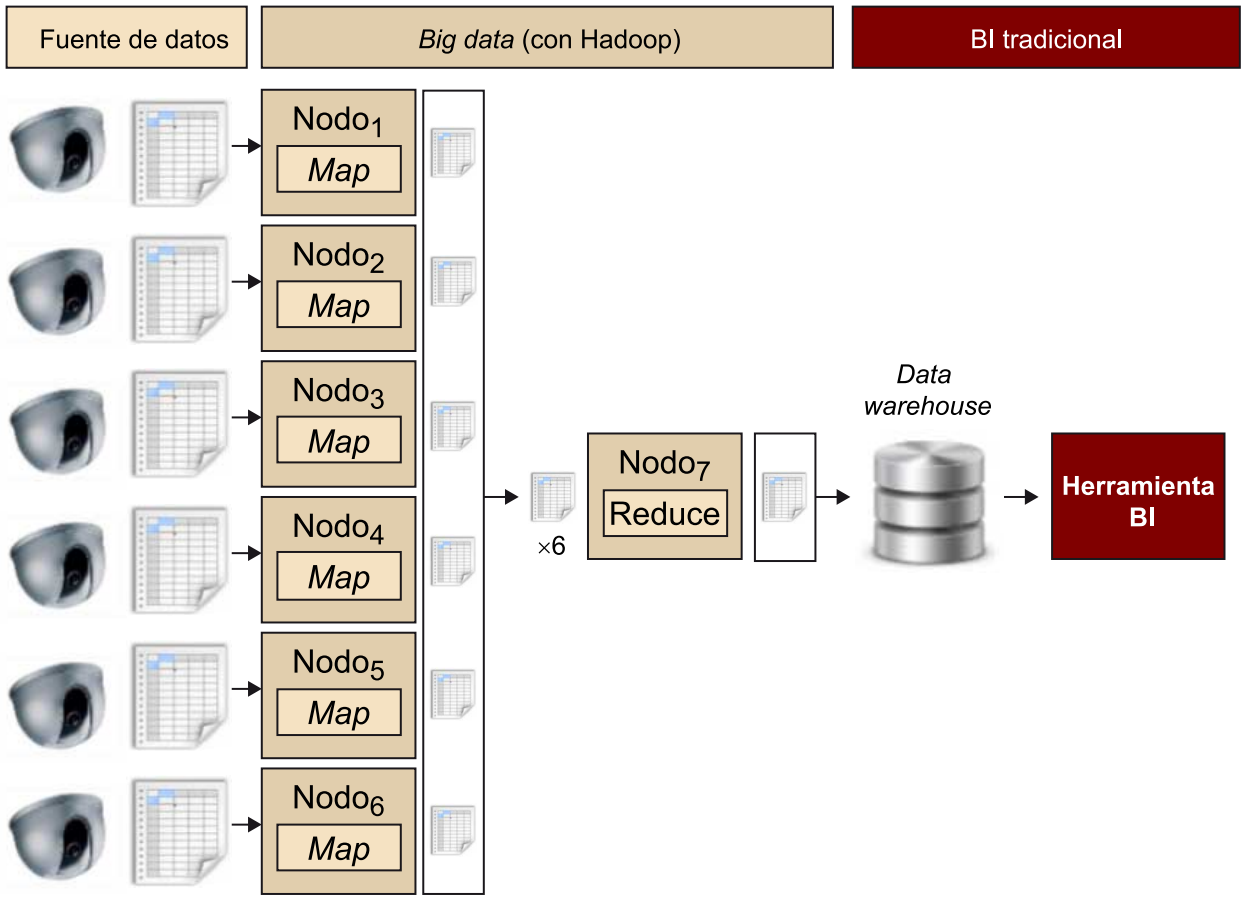
- Integración con software corporativo de IBM.

- Mejoras en la plataforma y el rendimiento.

3.1.5. Ejemplo de escenario *big data*

El siguiente diagrama muestra un ejemplo muy simplificado de proceso de datos desde su generación en una fuente de datos hasta su almacenamiento en un *data warehouse*.

Ejemplo de escenario *big data*



En este escenario basado en el ejemplo de un museo con un gran número de sensores de vigilancia por detección de movimiento y localización de personal de seguridad, se puede observar que los sensores generan grandes volúmenes de datos. Estos datos son almacenados en nodos del clúster de Hadoop[4] donde se aplicará un programa de *map*. Los resultados de la ejecución de estos programas son entonces enviados a otro nodo que ejecuta el programa *reduce* sobre los datos recibidos de los distintos nodos, obteniendo la información deseada de los sensores de vigilancia. Una vez se han convertido los datos en información válida para el usuario, esta se almacena en el *data warehouse* para su posterior análisis mediante una herramienta de BI.

[(4)]En este módulo no se detalla cómo los datos son almacenados en los diferentes nodos del clúster de Hadoop por tratarse de un tema claramente técnico. Este tema se tratará durante el segundo año del máster.

Con esta solución, el usuario puede analizar la información proveniente de los sensores a partir de un tratamiento previo realizado sobre los datos (programas MapReduce). Este tratamiento permite la reducción del volumen de datos a analizar y la fácil interpretación por parte del usuario, que se evita acceder a toda la información disponible y en un formato que puede que no sea útil para su análisis.

Durante el proceso, los datos generados por los sensores son almacenados en una base de datos NoSQL con capacidad para tratar datos complejos. Los nodos obtienen los datos a tratar de la base de datos NoSQL y a su vez dejan los resultados de la operación de *map* en esta base de datos para obtener persisten-

cia en los mismos. El nodo encargado de la operación *reduce* obtiene entonces esos datos, los analiza y finalmente los deja disponibles en el *data warehouse*, desde donde serán consultados a través de la herramienta de BI.

4. Una nueva figura en el equipo: El *data scientist*

Big data significa el paso a un nuevo nivel de análisis, donde, virtualmente, todos los datos generados en un sistema informático o dispositivo son susceptibles de análisis.

Para poder utilizar estos datos y obtener información útil para las organizaciones, es necesario la existencia de una figura capaz de:

- entender el origen y significado de los datos,
- extraer información útil de esos datos, y
- comunicar los beneficios del uso de esa información dentro de la organización.

Esa figura emergente dentro del área de *big data* es el llamado **científico de datos**, conocido habitualmente por su nombre en inglés: ***data scientist***.

4.1. Responsabilidades del *data scientist*

4.1.1. Entender los datos

Los analistas de datos poseen conocimientos en un área de negocio que les proporciona la capacidad de interpretar lo que los datos significan.

Un *data scientist* tiene un perfil de analista de datos, ya que posee los conocimientos necesarios para entender los datos. Conoce el área de negocio y sabe relacionar esos datos con la realidad que representan, más allá de simples datos aislados sin ningún significado en su totalidad. Es decir, puede ver los datos dentro de su contexto.

Por ejemplo, un fichero con los geoposicionamientos del personal de seguridad de un museo durante sus paseos nocturnos puede ser leído por cualquiera como un conjunto de coordenadas en el tiempo. Sin embargo, esos datos, en manos de una persona que entienda los datos y los sepa interpretar, suponen recorridos a través de las estancias, lo que conlleva la cobertura de estas por personal de seguridad y la desatención de otras estancias durante cierto periodo de tiempo. Más adelante veremos cómo un *data scientist* puede extraer información útil de estos datos.

Otro ejemplo es el de los datos generados por los sensores de movimiento, donde ciertas alarmas pueden ser el resultado de los paseos del personal de seguridad o de un intruso (a este tipo de alarmas de movimiento las vamos a llamar alarmas de tipo “persona”), o de insectos revoloteando o posándose

sobre el sensor (alarmas de tipo "insecto"). Dichas alarmas deben ser interpretadas para poder obtener información útil de esos datos, ya que una alarma de tipo "insecto" no debería generar ningún tipo de acción de seguridad, mientras que una alarma de tipo "persona" sí que debería, ya que podría tratarse de un intruso.

En la siguiente tabla se pueden apreciar tres datos referentes a tres tipos diferentes de alarma:

Id Evento	Id Sensor	Fecha	Coord. X	Coord. Y	Coord. Z	Volumen (cm^3)
1	FE032	23/05/2012 04:17:24.376	32,53	17,11	7,68	4
2	FR291	17/06/2012 03:54:15.577	15,43	17,44	0,92	78945
3	JK934	20/12/2012 04:12:55.412	23,22	9,21	0,11	5412

Una persona experta en la interpretación de los datos y su contexto sería capaz de inferir la siguiente información para cada uno de los registros:

Id Evento	Planta	Distancia al suelo (m)	Estancia	Elemento
1	2	3,52	Sala 234	Insecto (posible cucaracha)
2	1	0,92	Sala 114	Persona
3	1	0,11	Sala 130	Cuerpo pequeño (gato, perro, pájaro, robot?)

4.1.2. Extracción de información útil

Un dato por sí solo puede proporcionar información útil. Sin embargo, es el conjunto de los datos lo que suele proporcionar la información más valiosa. La combinación de datos de diferente origen y la comparación de datos similares es una gran fuente de información.

Las herramientas de BI son las grandes aliadas de los analistas de datos ya que les proporcionan una interfaz de usuario con un modelo de datos basado en el modelo de negocio que ellos conocen y dominan a la perfección. Gracias a esta interfaz, los analistas son capaces de generar análisis de gran complejidad basados en los datos almacenados. Sin embargo, los analistas de negocio suelen ser incapaces de llevar a cabo esta tarea sin una herramienta que les facilite el acceso a los datos.

Un *data scientist* tiene los conocimientos técnicos suficientes como para adentrarse en los datos sin ninguna interfaz de usuario previamente adaptada para el análisis de información. En otras palabras, es capaz de generar consultas a bases de datos, extraer datos de ficheros y programar rutinas de extracción de datos que le permiten "jugar" con los datos. De esta manera, es capaz de combinar datos de distintos orígenes, de agruparlos y de compararlos para encontrar patrones de comportamiento, tendencias y relaciones entre los datos.

Esta capacidad de poder manejar los datos es eminentemente técnica. No es suficiente el dominio de una herramienta de manejo de datos como puede ser una hoja de cálculo, ya que la extracción de datos requiere unos conocimientos adicionales. Además, hay que recordar que estamos tratando con grandes volúmenes de datos, lo que limita el uso de las hojas de cálculo, tanto por capacidad como por rendimiento.

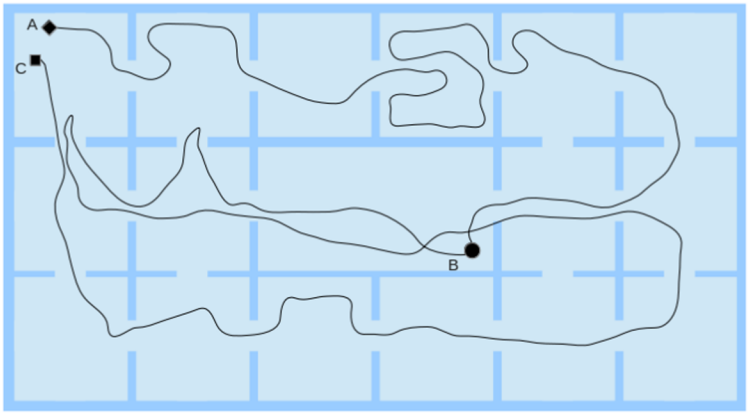
Si bien es cierto que un desarrollador podría extraer los datos y transformarlos en información, este carece del conocimiento suficiente de los datos como para poder indagar en ellos de manera independiente. Siempre necesitaría de alguien que le dirigiera.

En el ejemplo del geoposicionamiento del personal de seguridad del museo, los datos podrían relacionarse para dibujar el recorrido de las guardias nocturnas. El proceso por el cual los datos individuales pasarían a convertirse en un recorrido requiere de ciertos conocimientos técnicos que, seguramente, no estarían disponibles en un analista de negocio. Una manera de almacenar esos recorridos podría ser mediante sucesiones de puntos basados en cuatro dimensiones (coordenadas X, Y y Z, y el instante de tiempo).

Si entonces cruzamos los diferentes recorridos (mediante técnicas de resolución de sistemas de ecuaciones y análisis numérico), podríamos obtener los puntos de cruce de las guardias, o los momentos en que los guardias se concentran mucho en una zona y dejan otras desiertas. Esta información sería muy útil ya que permitiría realizar cambios en los recorridos con tal de cubrir una mayor superficie del museo con el personal de vigilancia existente.

El siguiente gráfico muestra el recorrido del vigilante 1.

Recorrido del vigilante 1



Sobre el papel, podemos ver lo que podrían ser anomalías en lo que se refiere a un recorrido estándar por todas las estancias. Se puede apreciar cómo en la cuarta estancia del recorrido parece haber un recorrido por los diferentes cuadros del museo en lugar de una inspección de la estancia. Esta información ya es valiosa por sí misma.

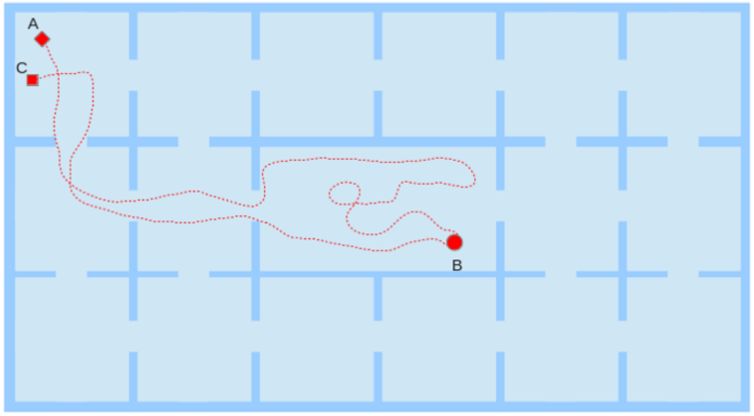
Observemos ahora la variación de posición respecto al tiempo. Esta información se muestra en la siguiente tabla:

Origen	Destino	Tiempo (minutos)
A	B	5
B	B	10
B	C	3

Con esta información podemos deducir que el vigilante estuvo detenido en el punto B durante 10'. Esta información es aún más valiosa.

Ahora observemos el recorrido del vigilante 2.

Recorrido del vigilante 2



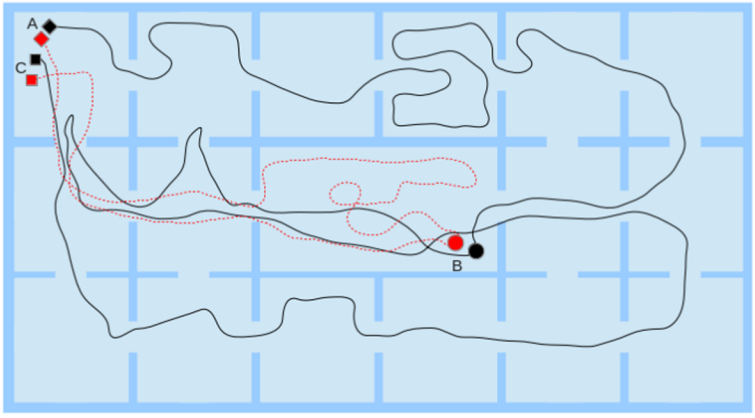
Y añadamos los intervalos de tiempo entre los puntos marcados en el gráfico:

Origen	Destino	Tiempo (minutos)
A	B	3
B	B	15
B	C	1

Se observa que el vigilante 2 estuvo detenido en ese punto durante la mayor parte del recorrido.

Finalmente, combinamos los dos gráficos:

Recorrido de los vigilantes 1 y 2



Teniendo en cuenta los tiempos entre puntos, podemos observar cómo el vigilante 1 y el vigilante 2 han coincidido en el punto B desde el minuto 5 hasta el minuto 15. Es decir, 10' sobre un total de 18' de recorrido.

Puede ser que ese sea su recorrido estipulado y que lo ejecuten a la perfección, pero esta información permite mostrar cómo otras estancias prácticamente carecen de vigilancia con esta planificación. Y esta información solo es posible gracias a la capacidad de extracción de información a partir de los datos que nos ha proporcionado el *data scientist*.

En el ejemplo donde se detectaban alarmas de movimiento, la combinación de dichas alarmas con el posicionamiento de los vigilantes permitiría descartar la alarma como un peligro para la seguridad del museo. En cambio, si al combinar ambos datos no hubiese ningún vigilante en ese punto y el tipo de alarma fuese de persona, debería activarse la alarma antirrobo.

4.1.3. Comunicar los beneficios a los usuarios

Uno de los grandes retos de *big data* es exponer una información que hasta el momento había sido descartada para el análisis debido a la imposibilidad de extracción de información mediante las técnicas tradicionales de BI.

El hecho de tratarse de fuentes de datos que los usuarios consideren como no fiables o de calidad discutible añade una dificultad a la hora de convencer a los usuarios de los beneficios que pueden obtener a partir de esta nueva información disponible en la solución BI corporativa. Es por eso por lo que el *data scientist* debe ser un gran comunicador.

Sin embargo, no hay suficiente con ser un buen comunicador. Los usuarios o patrocinadores del proyecto seguramente pondrán en duda algunas de las conclusiones, los beneficios o la manera de obtener la información, posiblemente con comentarios muy ligados a los mismos datos y la manera de interpretarlos y obtenerlos. Llegados a ese nivel de detalle, el jefe de proyecto o el responsable del área de negocio no podrán justificar las conclusiones plenamente, por lo que la persona idónea para esa tarea es el *data scientist*.

Así pues, volviendo a los ejemplos del museo, sería el *data scientist* quien se reuniría con los responsables de seguridad del museo y con los patrocinadores del proyecto BI para informarles de la posibilidad de utilizar los datos generados por los sensores de movimiento y posicionamiento. Durante esa reunión les informaría de los beneficios resultantes de su uso, como por ejemplo:

- Permitir la simplificación y automatización de los procesos de detección de alarmas.

- Optimizar los recorridos para maximizar la cobertura de seguridad en las estancias del museo.

- Detectar y corregir rutas anómalas en los recorridos de los vigilantes nocturnos.

- Monitorizar en tiempo real el posicionamiento del personal de seguridad para la gestión eficiente de alarmas.

4.2. *Data scientist* frente a Equipo de proyecto BI tradicional

A primera vista puede parecer que ese rol podría existir ya dentro del equipo de proyecto BI. Sin embargo, tal y como ya se ha indicado, no existe ningún rol en un equipo de proyecto BI tradicional que posea las capacidades y conocimientos del *data scientist*.

De hecho, un *data scientist* aglutina las características de varios roles establecidos de un proyecto BI. La siguiente tabla resume estas capacidades y hace un paralelismo entre estas y los diferentes roles dentro de un equipo de proyecto BI tradicional.

Capacidad	Rol en un equipo de proyecto BI
Entender los datos	Analista de negocio Arquitecto de datos Responsable de información Arquitecto ETL Arquitecto BI
Extraer información útil	Desarrollador ETL Desarrollador BI
Comunicar los beneficios a los usuarios	Jefe de proyecto Responsable del área de negocio

De la tabla anterior se desprende que no existe ningún rol en un equipo de proyecto BI tradicional con las capacidades necesarias para realizar las tareas asignadas a un *data scientist*. De aquí la necesidad de incluir en el equipo de proyecto BI con *big data* este nuevo rol.

5. *Big data* y el *cloud*

La implementación de una solución *big data* en el *cloud* ofrece importantes beneficios respecto a un sistema basado en un centro de proceso de datos local, debido a la flexibilidad que aporta el *cloud*.

En este apartado vamos a profundizar acerca de:

- el *cloud* y sus características, y
- los beneficios e inconvenientes del *cloud* respecto a una solución local.

5.1. Qué es el *cloud*?

El término *cloud* se suele utilizar con dos acepciones diferentes, aunque muy similares, cosa que puede llevar a la confusión. Si dichas acepciones existieran en un diccionario, veríamos algo similar a lo siguiente:

***cloud*[1]: infraestructura**. Dícese del hardware ubicado en algún centro de proceso de datos situado fuera del ámbito local de una organización. El hecho de contratar infraestructura en el *cloud* se conoce también como *hosting*.

Ejemplo

Una organización tiene un centro de proceso de datos limitado en espacio y capacidad. En un momento dado se plantean la introducción de un sistema de BI. Debido a la imposibilidad de ampliar su centro de proceso de datos local, se decide contratar servidores externos para albergar los sistemas informáticos necesarios para la nueva solución BI.

***cloud*[2]: servicios**. Dícese del conjunto de servicios ofrecidos por proveedores externos, que integran tanto hardware como software, y que están ubicados fuera del ámbito local de una organización. Estos servicios incluyen multitud de sistemas informáticos, como por ejemplo bases de datos, servidores web, servidores de aplicaciones, capacidad de almacenamiento y servidores de correo, entre otros muchos. Este concepto se conoce también como *cloud computing*.

Ejemplo

Una organización desea implantar un sistema BI. Para ello necesita un conjunto de servidores y un software del que carece en estos momentos (base de datos para el *data warehouse*, herramienta de ETL y herramienta de BI). Para ello, en vez de comprar nuevos servidores y adquirir el software necesario, se decide contratarlos a una empresa que ofrece servicios de *cloud computing*, de manera que el proveedor de servicios se ocupa del mantenimiento tanto de la infraestructura como de las aplicaciones con un coste fijo anual.

En este apartado nos referimos al *cloud* como ***cloud computing***. Es decir, nos referimos a la provisión de hardware y software por un proveedor externo, localizado fuera del ámbito local de una organización.

5.2. Beneficios e inconvenientes respecto a una solución local

Los beneficios que aporta el *cloud* respecto a una solución local son:

- Flexibilidad, adaptabilidad y coste ajustado al uso del sistema.
- Eliminación de mantenimiento interno del sistema, tanto en lo que se refiere a hardware como a software.

Por otra parte, las desventajas que ofrece son:

- Dependencia de una conexión externa.
- Menor control de la plataforma.

A continuación se expanden estos beneficios e inconvenientes, primero de manera genérica y después mediante un ejemplo basado en el uso del *cloud* con *big data*.

5.2.1. Beneficio: Flexibilidad y coste ajustado al uso del sistema

Si tenemos en cuenta la necesidad de almacenamiento en un entorno BI, vemos que el volumen necesario crece muy rápidamente. Como en todo sistema informático, a la hora de proveer un sistema BI con *big data* de espacio de almacenamiento necesario para ubicar todos los datos, será necesario:

- una estimación del volumen de los datos al iniciar el sistema;
- una estimación de crecimiento en el tiempo;
- un cálculo del volumen inicial de almacenamiento deseado, con vistas a una ampliación después de cierto tiempo o cuando el volumen real se acerque al volumen total disponible.

En base a esto, la organización deberá adquirir espacio en disco inicialmente, para luego ampliarlo sucesivamente. De ello se derivan los siguientes problemas:

- Se produce un gasto elevado inicial cuando, en realidad, no se está utilizando toda la capacidad adquirida. Esta capacidad de recursos puede ser con respecto a:
  - Espacio de almacenamiento,
  - CPU,
  - memoria RAM,

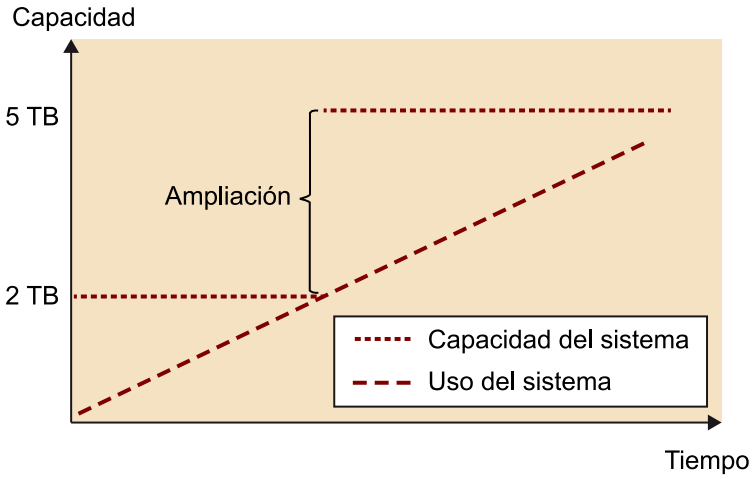
– ancho de banda.

- La máxima capacidad de los recursos adquiridos se produce justo antes de la necesidad de ampliación.

- Cada ampliación del sistema por lo que se refiere al hardware supone una intervención en el sistema con un más que posible paro de la disponibilidad del sistema.

El siguiente gráfico muestra cómo, al llegar a un nivel crítico de capacidad en un sistema, este debe ser ampliado para poder continuar creciendo y dando soporte a la solución BI.

Capacidad de almacenamiento y ampliaciones a través del tiempo

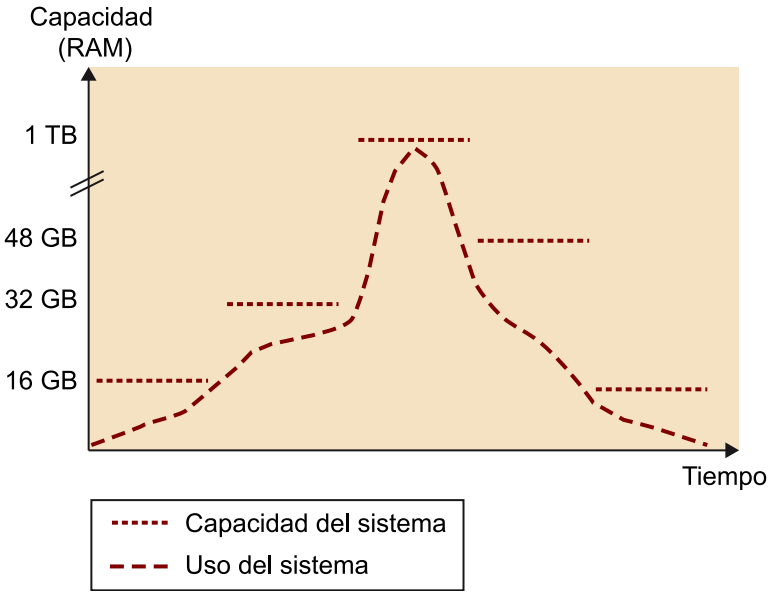


Lo que se ha descrito hasta ahora es algo común a todo sistema informático. Sin embargo, trabajar en el *cloud* nos ofrece más flexibilidad en esta área.

Los servicios en el *cloud* permiten la ampliación de los recursos de manera transparente mediante la técnica de la virtualización, de manera que ya no es necesario realizar una intervención en el sistema para ampliar los recursos de este. Además, esta ampliación puede hacerse de manera dinámica en función del consumo de recursos, lo cual minimiza el valor de los recursos no utilizados en el sistema, reduciendo el coste total de este. Esto se traduce en ampliaciones o reducciones de la capacidad según lo estipulado por ciertas reglas definidas por el usuario, tanto para la ampliación como para la reducción de recursos. Por ejemplo, "cuando el consumo de memoria RAM esté por encima del 95%, ampliar el sistema con un 20% más", o "cuando el uso de memoria RAM esté por debajo del 40% durante más de 5', reducir la capacidad un 50%").

En el siguiente gráfico se puede observar cómo la capacidad del sistema se modifica en función del uso de los recursos.

Flexibilidad de recursos



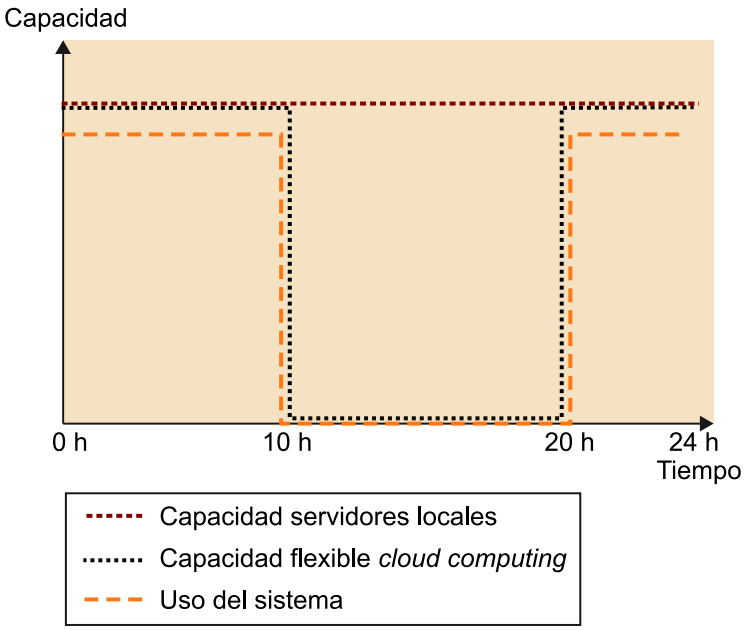
Ejemplo: Obtención de datos de sensores

Revisemos el caso de un museo con sensores de detección de movimiento en las diferentes estancias y geoposicionamiento del personal de seguridad. Supongamos que dichos sensores se activan al cerrar el horario de visitas. En ese caso, veremos que dentro de un ciclo de 24 horas se produce una gran actividad en el sistema al activar los sensores (por ejemplo, las 20:00) y se paraliza dicha actividad al empezar la jornada en horario al público (por ejemplo, las 10:00).

Si hemos optado por una solución implementada en el centro de proceso de datos del museo con servidores físicos, el museo deberá adquirir servidores que puedan soportar la gran demanda derivada de la solución BI con *big data*, pero tan solo utilizará el potencial de esos servidores durante un periodo de 14 horas. En cambio, si hemos optado por una solución basada en *cloud computing*, podremos dimensionar la capacidad de los servidores en función de la necesidad real de los recursos del sistema. En este caso, podremos tener un clúster de nodos de Hadoop para tratar los datos de los sensores del museo en el periodo de operatividad de estos (de 20:00 a 10:00), mientras que fuera de este periodo, podemos tener un dimensionamiento mínimo para, por ejemplo, extraer diferentes estadísticas del sistema. En este caso de recogida de estadísticas durante el día, no habrá que procesar grandes volúmenes de datos en tiempo real para detectar posibles intrusos en el museo, por lo que la capacidad del sistema no requerirá de la capacidad de un sistema BI con *big data*.

Podemos representar la actividad del sistema y la capacidad de este en ambos casos mediante el siguiente gráfico:

Actividad en el sistema de sensores en un ciclo de 24 horas



Tal y como se puede apreciar en este gráfico, el sistema, con *cloud computing*, se puede dimensionar de manera acorde al uso que se hace de este.

5.2.2. Beneficio: Eliminación de mantenimiento del sistema

En un sistema informático basado en *cloud computing*, el mantenimiento del centro de proceso de datos, del hardware y del software, no son competencia de la organización que paga por esos servicios (el cliente), sino de quien los ofrece.

Esto permite al cliente liberarse, entre otras, de las siguientes tareas:

- Ampliaciones y reducciones del hardware usado.
- Actualizaciones de software.
- Reemplazo de componentes hardware.
- Mantenimiento y acondicionamiento del centro de proceso de datos.
- Formación y actualización del personal encargado del mantenimiento tanto del hardware como del software.
- Elaboración y ejecución de planes de recuperación en caso de desastre.

Gracias a esto, las organizaciones pueden centrar sus esfuerzos en otras tareas más en línea con sus objetivos.

En el caso de un proyecto BI con *big data*, la flexibilidad que ofrece trabajar con servicios en el *cloud* es sinónimo de múltiples ampliaciones y reducciones de hardware, de (relativamente) frecuentes actualizaciones de software para adquirir las nuevas versiones de los componentes del sistema[5], y de formación del personal de sistemas para poder gestionar este software. Con todo esto, es lógico pensar en elevados costes de mantenimiento (explotación y sistemas) del entorno *big data*.

[5]Es importante recordar que la tecnología usada en proyectos *big data* es aún muy joven, lo que conlleva ciclos relativamente cortos de desarrollo de software por parte de los fabricantes de dicho software.

Ejemplo: Prueba piloto de implantación de una solución *big data*

Supongamos que el museo de nuestro ejemplo desea hacer una puesta en marcha restringida y limitada a algunas funcionalidades para poder probar la solución en pequeña escala y decidir posteriormente si se procederá al despliegue en todo el museo.

Para esa prueba piloto no será necesario disponer de un hardware a gran escala para tratar la información de los centenares de sensores. Asimismo, tampoco se necesitará instalar y configurar todo el software, ya que algunas de las funcionalidades del sistema final no serán utilizadas.

Si se utilizase un sistema tradicional, se necesitaría un equipo con conocimientos técnicos para la instalación y configuración del hardware y el software para la prueba piloto, aun sin saber si el proyecto se acabará realizando o no. En este caso, la formación de personal propio es una inversión con riesgo, cara y lenta; la incorporación de personal nuevo en plantilla introduce un elevado riesgo; y la consultoría puede ser demasiado cara, no garantiza la mejor calidad del servicio y crea una dependencia de personal externo.

En cambio, la contratación de servicios en el *cloud* permite de manera simple la gestión de la plataforma por personal especializado y con una amplia experiencia en la gestión y el mantenimiento de esos componentes informáticos. Gracias a los servicios en el *cloud*, el museo es capaz de tener una plataforma funcional preparada para el desarrollo de la prueba piloto y otra para sus pruebas de manera rápida. Nadie en el departamento de IT se ha tenido que formar en la instalación, configuración y verificación de ninguno de los sistemas puestos en marcha en el *cloud*. Y si el proyecto se lleva a cabo, cuando se necesite más hardware, tan solo habrá que pedir dicha ampliación en una consola de servicios en el *cloud*, en lugar de tener que dedicar recursos propios.

5.2.3. Inconveniente: Dependencia de una línea externa

Al tener parte de la infraestructura necesaria para el correcto funcionamiento de un sistema informático en una localización remota, se produce una dependencia de una conexión permanente con el *cloud*.

En los casos de *big data*, donde existen grandes volúmenes de datos y la necesidad de altas velocidades de proceso de datos, se requiere típicamente una conexión con un gran ancho de banda y una gran fiabilidad.

Existen diversas formas de conexión al *cloud*. La más habitual es mediante una conexión a la red Internet. Para poder garantizar una conexión estable, es habitual en organizaciones la existencia de diversas líneas de acceso a la red Internet con distintos proveedores para poder conectarse con una de ellas (línea o conexión de *backup*) en caso que la principal deje de funcionar. Por su parte, los proveedores de servicios de *cloud computing* suelen ofrecer conexiones mediante redes privadas que permiten la conexión sin depender del proveedor de acceso a la red Internet.

En ambos casos se produce un coste adicional para poder conectar con un sistema informático externo. Algo que, en el caso de tener la infraestructura en un centro de proceso de datos local, no sería necesario.

Ejemplo: Despliegue de la solución *big data* en modo producción

En el museo han decidido implantar la solución *big data* mediante servicios en el *cloud*. La conexión se realizará mediante líneas de alta velocidad convencionales contratadas a un ISP[6]. Esto supone la transmisión de los datos generados por cientos de sensores a través de la red Internet. La criticidad de los datos para la seguridad del museo (los datos deben tratarse con el mínimo retraso) recomienda la utilización de líneas de *backup* por si la comunicación entre el museo y el proveedor de servicios en el *cloud* a través de las líneas principales dejase de funcionar. Para evitar dependencias con un único ISP, se contratan las líneas de *backup* a un ISP alternativo.

(6)Internet Service Provider (proveedor de servicios de Internet)

5.2.4. Inconveniente: Menor control de la plataforma

Al contratar servicios de *cloud computing*, se pierde parte del control sobre la plataforma en lo que se refiere al hardware.

Cuando se contrata un clúster de servidores con 32 CPU, 256 GB RAM y 8 TB de almacenamiento en disco, se desconoce por ejemplo su ubicación real, quién tiene acceso a ese clúster y los mecanismos de prevención de desastres (por ejemplo, incendios), entre otras cosas. Esta situación puede provocar ciertos recelos o inseguridad en el cliente debido a esa falta de control.

Además, no existe un control sobre el hardware como en un centro de proceso de datos local. Por ejemplo, trabajando con infraestructura local es relativamente fácil cambiar una cabina de discos por otra más rápida. En cambio, esa capacidad puede que no esté disponible para su contratación en un proveedor de servicios *cloud computing*.

En algunas organizaciones, el inconveniente de falta de control podría incluso llegar a bloquear la opción de trabajar en el *cloud*. Aquí es donde las políticas internas de las organizaciones sobre seguridad y control de sus entornos pueden jugar un papel decisivo (en el caso de veto), o no.

Ejemplo: Incorporación de datos personales de los vigilantes a la solución

En el museo han decidido añadir información sobre los vigilantes al sistema BI con *big data* para así poder obtener mejores conclusiones durante las tareas de análisis. Se incluyen datos como el nombre y apellidos, la dirección de residencia, el número de teléfono privado, información familiar, historial laboral y antecedentes penales.

Este requerimiento provoca la reacción del departamento legal, ya que según la normativa vigente en el país donde se halla el museo, los datos personales deben estar protegidos mediante una rigurosa serie de medidas de seguridad.

El departamento de IT asegura al departamento legal que los datos están seguros en el *cloud* y que cumplen con todas las normativas de seguridad vigentes, a pesar de estar situados en un centro de proceso de datos situado en un país extranjero. A este último punto, el departamento legal responde con objeciones y recomienda el uso de un proveedor de servicios en el *cloud* con un centro de proceso de datos situado en el territorio del país donde se encuentra el museo.

En este caso, hay dos opciones abiertas: cambiar de proveedor o rebajar el nivel de exigencia legal. En un caso real, se deberá llegar a una situación de acuerdo entre ambas partes para poder desatascar la situación.

6. Métricas de evaluación

6.1. Análisis de coste (TCO)

Tal y como se ha indicado anteriormente, en una solución BI con *big data*, añadiremos componentes tecnológicos sobre la base de una solución BI tradicional. Partiendo de esta base, en este apartado nos centraremos tan solo en el estudio del TCO para los componentes específicos de una solución BI con *big data*. Este coste deberá ser añadido al TCO de una solución BI tradicional.

Hay que tener en cuenta que los componentes tecnológicos de una solución BI con *big data* van a depender de las necesidades específicas de cada solución. Debido a la disparidad de escenarios agrupados bajo el término *big data* (grandes volúmenes, alta velocidad de tratamiento de datos, variedad de datos), los componentes pueden llegar a ser muy variados y específicos.

En la siguiente tabla se muestran estos componentes para un sistema BI tradicional y los tres componentes básicos más utilizados en una solución BI con *big data*.

Componente	BI tradicional	*Big data*
Sistema de ficheros de datos distribuido		X
Base de datos NoSQL		X
Hadoop		X
Sistema gestor de bases de datos (SGBD) para el *data warehouse*	X	X
Herramienta BI	X	X

6.1.1. Consideraciones en el cálculo del TCO

La solución tecnológica determina los componentes del sistema. El TCO debe incluir el coste de esos componentes a nivel de licencias de producto y mantenimiento.

Llegados a este punto, cabe recordar que existen soluciones tecnológicas *open source* para todos los componentes de *big data*, así como productos licenciables de fabricantes de software.

Hay que tener en cuenta todo el coste derivado de la utilización de software y el hardware requerido. A continuación se muestra una lista de áreas donde podemos incurrir en coste para cada uno de ellos:

- Hardware:
  - Servidores donde se instalará el software. Hay que tener en cuenta los múltiples entornos que podemos tener: desarrollo, integración, producción…
  - Tiempo de dimensionamiento, búsqueda y compra de servidores.
  - Tiempo de instalación y configuración de servidores.
  - Posibles ampliaciones futuras.

- Software:
  - Licencias de las herramientas citadas anteriormente.
  - Licencias de otro software necesario. Esto incluye el sistema operativo.
  - Licencias de soporte y mantenimiento para años venideros.
  - Instalación de software
  - Instalación de futuros *upgrades* y *patches*.

- Migración:
  - Migraciones de software si el software actual no dispone de las capacidades deseadas y se requiere un cambio de herramienta BI (por ejemplo).
  - Servidores donde realizar la migración.
  - Desarrollo de proyecto de migración.

- Mitigación de riesgos:

- Desarrollo de la solución BI con *big data:*
  - Proyecto completo.
  - Formación de equipo.
  - Coste de contratación de servicios o personal, de manera opcional.
  - Formación al departamento de explotación.
  - Coste de personal del equipo del proyecto y otros recursos (*data scientist, management*, personal de negocio, etc.).

Como ya se ha comentado en el apartado anterior, la introducción de *cloud computing* provoca un cambio en el modelo del TCO. Esta es una opción a tener muy en cuenta en proyectos de *big data*. En este caso, el modelo de coste se basa típicamente en:

- Dimensionamiento de hardware basado en los siguientes recursos:
  - espacio de almacenamiento,
  - CPU,
  - memoria RAM,
  - ancho de banda.

- Número y tipo de licencias de los paquetes de software, como por ejemplo:
  - usuarios de base de datos,
  - usuarios de herramienta BI,
  - opciones licenciables de base de datos, módulos Hadoop y de herramienta BI.

Conviene ser muy estrictos en el cálculo del TCO para poder así calcular con precisión el retorno de inversión y el periodo de *payback*.

6.2. Análisis de retorno de inversión y periodo de *payback* (*ROI* & *payback period*)

Empecemos haciendo un repaso a las definiciones de *ROI* y *payback period*:

1) *Return on investment* (*ROI*)

El *ROI* se define como la eficiencia de una inversión en función del gasto realizado. El *ROI* se expresa como un porcentaje y se calcula mediante la siguiente fórmula:

$$ROI\% = \frac{\text{Beneficios derivados de la inversión} - \text{Coste de la inversión}}{\text{Coste de la inversión}} \times 100$$

Ejemplo:

Beneficios derivados de la inversión	100.000 €
Gastos de inversión	40.000 €

$$ROI\% = \frac{100.000\,€ - 40.000\,€}{40.000\,€} \times 100 = 150\%$$

2) *Payback period*

El periodo de *payback* se define como el periodo de tiempo necesario para recuperar la inversión realizada en un proyecto. Se expresa en unidades de tiempo y se calcula mediante la siguiente fórmula:

$$Payback\ period = \frac{\text{Gastos de inversión}}{\text{Beneficio neto de inversión por unidad de tiempo}}$$

Ejemplo:

Gastos de inversión	40.000 €
Beneficio neto de inversión por unidad de tiempo	5.000 € / 2 meses

$$Payback\ Period = \frac{40.000\ €}{5.000\ €/2\ \text{meses}} = 16\ \text{meses}$$

6.2.1. Consejos para calcular el *ROI* y el *payback period*

A pesar de disponer de una fórmula simple para el cálculo del *ROI* y el *payback period*, el problema con el que nos podemos topar en este momento es el de saber qué valores vamos a introducir en cada caso.

Estos consejos van dirigidos a obtener las variables de las fórmulas de *ROI* y *payback period*.

1) Beneficios

Como beneficio se entiende la cuantificación en la unidad de medida moneda que aporta la solución de BI. Para poder cuantificar estos beneficios, debemos encontrar maneras de traducir los beneficios no tangibles a esa unidad de medida moneda.

A continuación se muestran varios ejemplos de cómo calcular beneficios en soluciones de BI en tiempo real.

Ejemplo 1: Tiempo ahorrado en la búsqueda y análisis de la información

Big data nos permite analizar los datos para obtener información útil en un periodo de tiempo inferior al que invertiríamos al utilizar un sistema BI tradicional. Si un recurso se encuentra parado, esperando una respuesta del sistema BI, estamos malgastando ese recurso. Esa diferencia de tiempo puede expresarse en términos de beneficio mediante la siguiente fórmula:

$$\text{Beneficio} = (\text{Tiempo BI tradicional} - \text{Tiempo Big Data}) \times \text{Coste horario recurso} \times \\ \times\ \text{Núm. de veces en el periodo}$$

Ejemplo 2: Reducción de costes por liberación de recursos

En el caso del museo que hemos estado viendo a lo largo de este módulo, el análisis de recorridos de vigilantes puede concluir en un rediseño de dichos recorridos para obtener una mayor eficiencia. Esto puede desencadenar una reducción en el número de recursos utilizados, con la consiguiente reducción de gastos de personal ya sean subcontratados o en plantilla.

En el caso de subcontratación o de despido de recursos en plantilla, obtenemos la siguiente fórmula:

$$\text{Beneficio} = \text{Núm. de recursos liberados} \times \text{Coste por recurso}$$

En el caso de recursos en plantilla, dichos recursos también podrían dedicarse a otras tareas que aporten un beneficio, sin necesidad de reducir la plantilla.

Ejemplo 3: Reducción de costes por obtención de información

Con *big data* podemos obtener información de donde antes no podíamos. El caso del análisis de comentarios en las redes sociales para la obtención del sentimiento / grado de fidelización respecto a una marca es un claro ejemplo. Sin *big data* habría que obtener esa información mediante otros medios como las encuestas, con el consiguiente esfuerzo económico que conlleva realizar dichas encuestas.

$$\text{Beneficio} = \text{Coste de encuestas no realizadas}$$

2) Gastos de inversión

Los gastos de inversión son los derivados de la implantación de la solución de BI con *big data* y su mantenimiento.

La siguiente lista incluye varias partidas a tener en cuenta:

- Coste de consultoría (si existe)
- Coste de personal interno (IT)
- Coste de personal interno (no IT)
- Coste de compra de licencias (primer año)
- Coste de licencias de mantenimiento (años siguientes)
- Coste de hardware y mantenimiento
- Coste de formación

7. Mitos

En este apartado se desea desmentir algunos de los mitos más populares que existen sobre *big data*. Los mitos descritos a continuación se han clasificado como conceptuales y técnicos. Los mitos conceptuales se refieren a *big data* en general, mientras que los mitos técnicos se centran en los componentes tecnológicos utilizados en la implementación de una solución BI con *big data*.

7.1. Mitos conceptuales

1) A mayor volumen de datos, más valor para la organización

Un mayor volumen de datos no garantiza la obtención de mejor información y una mejor toma de decisiones.

> Lo que nos permite tomar buenas decisiones es la calidad de la información, no la cantidad de datos disponibles.

Es cierto que, en general, disponer de grandes cantidades de datos nos permite extraer información fiable debido al gran muestreo de los mismos, pero eso no implica que dichos datos sean buenos para la toma de decisiones. Además, grandes volúmenes de datos complican el análisis de estos desde el punto de vista tecnológico, ya que necesitamos más potencia de cálculo y, en algunos casos, técnicas y aproximaciones totalmente diferentes para extraer información útil para el análisis decisional.

2) *Big data* es útil para grandes volúmenes de datos solamente

La necesidad de procesar grandes cantidades de datos es, sin ninguna duda, una justificación para la implementación de *big data*. Sin embargo, esta necesidad no es suficiente por sí sola para iniciar un proyecto BI con *big data*. Un sistema BI tradicional puede ser capaz de procesar grandes cantidades de datos y extraer información útil para su análisis. Solamente en el caso de que el sistema BI tradicional no pueda proporcionar una solución a los requerimientos de la organización, deberemos implementar *big data*. Además, hay que tener en cuenta que, dentro de los factores que pueden decantar un proyecto hacia *big data*, también se hallan la heterogeneidad de los datos y la velocidad de los procesos de tratamiento de datos.

La balanza se inclinará hacia *big data* cuando debamos procesar tipos de datos complejos para los cuales los SGBDR tradicionales no están preparados (imágenes, audio, vídeo, localización geoespacial, *web logs*, documentos de texto, etcétera) y cuando debamos procesar los datos de forma rápida para generar información útil y válida (que no haya caducado).

3) *Big data* trata solamente con datos no estructurados

Big data puede procesar cualquier tipo de datos, siempre y cuando dispongamos de un módulo (un programa) que permita tratar ese tipo de información. La creencia de que *big data* trata solamente datos no estructurados tiene su origen en el hecho que los SGBDR tradicionales no tienen las herramientas suficientes para procesar estos tipos de datos. Debido a esta limitación, se ha asociado el tratamiento de tipos de datos no estructurados (por ejemplo texto libre) a *big data*. No obstante, en determinadas circunstancias también puede ser aconsejable utilizar *big data* para procesar tipos de datos complejos, o semiestructurados (como por ejemplo, entradas en un foro o en redes sociales).

4) *Big data* solo es válido para grandes organizaciones

El tamaño de una organización se mide típicamente, bien en número de empleados bien en volumen de ingresos o beneficios. *Big data* no entiende de estas métricas, sino de datos. Cualquier organización puede tener grandes volúmenes de datos, la necesidad de procesar datos a una gran velocidad, o tipos de datos complejos o carentes de una estructura bien definida que deben ser analizados. Ese es el escenario donde *big data* aporta valor, independientemente de la organización.

Si la implementación de *big data* satisface las necesidades de acceso a la información de una organización, entonces *big data* es apto para esa organización.

5) *Big data* aplica solo para redes sociales

Cualquier tipo de aplicación informática, dispositivo electrónico o, en general, cualquier sistema que genere datos que deban ser analizados es susceptible de utilizar *big data*. La realidad es que el alto volumen de datos en las redes sociales ha hecho que, históricamente, *big data* se utilice para analizar esos datos, para, por ejemplo, medir el sentimiento sobre una marca, sobre una campaña de marketing o sobre un evento específico.

Sin embargo, hay un sinfín de escenarios donde *big data* puede aplicarse que no tienen ninguna relación con el análisis de datos provenientes de redes sociales.

7.2. Mitos técnicos

1) *Big data* es un problema básicamente tecnológico

Big data, como conjunto de técnicas y tecnologías, tiene una marcada vertiente tecnológica. Esta nueva aproximación al BI requiere de un trabajo de adaptación tanto de los equipos técnicos como del software y hardware al entorno de *big data*. Sin embargo, a pesar del gran impacto que esto supone respecto a una solución BI tradicional, este no es el único problema a tener en cuenta en una implementación de *big data*. Uno de los mayores retos en un proyecto *big data* es el de saber cómo procesar los datos disponibles para poder extraer información útil para la organización. Sin esta información no es posible proporcionar un valor añadido a la organización, con lo cual el proyecto *big data* pasaría a ser un fracaso.

2) *Big data* y Hadoop son lo mismo

Big data es un concepto que se refiere a un conjunto de técnicas y tecnologías para la extracción de información a partir de ciertos datos. Hadoop es una implementación concreta del paradigma de computación distribuida. O sea, una solución técnica que puede utilizarse para implementar *big data*, pero no la única. En el mercado existen otras herramientas comerciales que realizan la misma función. Sin embargo, la popularidad de Hadoop ha motivado este mito.

3) Es necesario saber Hadoop para entender lo que *big data* puede aportar a una organización

Hadoop (o cualquier otra solución que permita la computación distribuida), es una parte muy importante de *big data*. Es cierto que es muy importante conocer Hadoop, las opciones que ofrece y cómo trabaja, para poder sacar el máximo partido de la plataforma con la que vamos a implementar *big data*. Sin embargo, no es crítico tener este conocimiento ni para poder entender *big data*, ni para poder identificar la información a obtener al final del proceso una vez los datos hayan sido procesados por la solución *big data* implementada. De hecho, para esto, lo que se necesita es un buen entendimiento de los datos y de la información que podemos extraer de estos.

4) NoSQL significa "sin SQL"

La abreviatura de base de datos *NoSQL* (nótese la primera letra mayúscula y la segunda minúscula) se refiere a base de datos *Not only SQL*, es decir, *No solo SQL*. Esto se debe a que este tipo de bases de datos (o contenedores de datos, si queremos distinguirlos de la bases de datos relacionales), incorporan módulos nativos para el proceso de tipos datos complejos. Ejemplos de estos tipos de datos complejos que han sido incorporados a bases de datos NoSQL son documentos, grafos y datos geoespaciales entre otros. Una base de datos NoSQL es un complemento a un SGBDR, ya que permite el proceso de tipos de datos que este no es capaz de procesar.

Grafos

Un grafo es una representación gráfica abstracta basada en nodos y aristas, utilizada para representar un dominio de discurso. Por ejemplo, en el caso de una empresa de transportes, los nodos podrían ser los puntos de entrega y recogida de paquetes, y las aristas, los desplazamientos entre dichos puntos de entrega y recogida. Los grafos pueden ser dirigidos, lo cual implica que las aristas entre nodos tendrán flechas de dirección, y pueden estar etiquetados con un valor, como por ejemplo el número de kilómetros entre nodos o el tiempo. En este caso, podríamos extraer información que nos permitiese identificar rutas óptimas para la recogida y entrega de paquetes. En el apartado de materiales del aula se pueden encontrar los materiales de la asignatura de Matemática discreta, con más información sobre qué son los grafos y cómo trabajar con ellos.

En ningún caso debe sustituirse un SGBDR por una base de datos NoSQL, ya que las funcionalidades de ambas son complementarias.

8. Casos de éxito

A continuación se muestran casos de éxito de BI en *big data* para ilustrar los beneficios que puede aportar una solución de este tipo.

8.1. Mejora de la gestión del tráfico

El Royal Institute of Techonology (KTH) es la universidad politécnica más importante de Suecia y una de las más importantes de Europa. La división de tráfico y logística del KTH se dedica a la investigación de nuevos modelos que mejoren los sistemas de transporte y trafico existentes.

Uno de sus proyectos se definió de la siguiente manera:

- Se obtendrían datos de una gran variedad de fuentes de datos, entre las cuales se hallaban los siguientes:
  - Dispositivos GPS situados en un gran número de vehículos. Estos dispositivos muestran información precisa de la localización de los diferentes vehículos.
  - Radares de tráfico situados en calles y carreteras. Con la información de los radares es posible obtener la velocidad del tráfico en esa posición y la densidad del tráfico (número de coches por unidad de tiempo).
  - Información de accidentes para identificar puntos de tráfico potencialmente críticos.
  - Información de obras existentes en las diferentes vías de transporte, para identificar vías con movilidad reducida respecto a la normalidad.
  - Información meteorológica, para identificar si hay precipitaciones, de qué tipo (lluvia, nieve, granizo), con qué intensidad (litros / m^2·hora), qué cantidad acumulada hay (por ejemplo, 5 cm de nieve), si hay niebla, la visibilidad (por ejemplo, 15 metros), etc.
- Se combinarían esos datos y se analizarían para poder establecer las condiciones en el tráfico actual, y para poder identificar la mejor ruta entre dos puntos de la ciudad en cada momento.

Para poder procesar todos estos datos y poder ofrecer resultados útiles para los usuarios del sistema, se utilizó *big data*. El uso de *big data* viene justificado por el hecho de que se daban condiciones favorables en las tres áreas que caracterizan un sistema *big data*: gran volumen de datos, alta velocidad para transformar los datos en información útil, y gran variedad de datos.

- Se procesaría un **gran volumen de datos**, obtenidos principalmente del gran número de dispositivos GPS y radares de tráfico. Hay que tener en cuenta que cada uno de estos dispositivos es capaz de generar múltiples mediciones por segundo, con lo que el volumen de información es muy elevado. En este proyecto, el volumen de datos era del orden de magnitud de gigabytes/segundo.

- Se necesitaría **obtener los resultados** del análisis de los datos (las mejores rutas entre dos coordenadas) **en un tiempo mínimo**, para poder mostrarlas a los conductores con tiempo suficiente como para poder tomar esas rutas. Es decir, los datos deben ser procesados y analizados con un retraso mínimo. El objetivo es ofrecer a los usuarios información sobre las mejores rutas en ese mismo instante y no las de unos minutos antes, ya que, debido a la variabilidad del tráfico, la efectividad de las rutas puede cambiar radicalmente en cuestión de minutos.

- Se dispondría de **datos complejos** que no podrían ser tratados de manera eficiente por un sistema gestor de base de datos relacional (SGBDR). El hecho de trabajar con datos de geoposicionamiento y la necesidad de calcular distancias basadas no solo en la posición sino también en las vías disponibles para ir de un punto A a otro punto B, precisa de algoritmos de enrutamiento que no están disponibles en los paquetes básicos de los SGBDR comerciales[7]. Además, este proyecto utiliza los datos actuales para generar análisis predictivo, lo cual requiere de cálculos muy complejos[8].

Desde el punto de vista tecnológico se utilizó una solución basada en *streams*, para la obtención de datos y su posterior almacenamiento. De esta manera se evitaba el retraso introducido por una carga de datos ejecutada cada cierto intervalo de tiempo.

Streams

La obtención de datos mediante *streams* se basa en la distribución de información a medida que esta genera. De esta manera se produce un flujo continuo de datos desde el sistema origen al sistema consumidor de datos. De ahí el término *stream* (flujo, en inglés).

El sistema se diseñó para alimentarse de la información obtenida. Esto permite la comparación de múltiples rutas alternativas entre dos puntos dadas unas condiciones de tráfico concretas. Al identificarse una ruta como la más eficiente entre dos puntos, el sistema la cataloga como tal, mostrándola a los conductores cuando dichas condiciones se repitan en el futuro. Es decir, las

(7)Si bien es cierto que los paquetes básicos de los SGBDR comerciales no incluyen este tipo de algoritmos, pero también es cierto que varios de ellos incluyen extensiones geográficas que permiten realizar operaciones con datos geográficos. Ejemplos de estas extensiones son Oracle Spatial y PostGIS.

(8)En el documento "Dynamic Model of Network with Real Time Traffic Information: Queue Equilibrium and Stability Analysis" puede observarse la complejidad de un modelo dinámico de cálculo de tráfico. La solución *big data* debe calcular rutas de tráfico sobre la base de una gran cantidad de información, teniendo en cuenta muchas variables. Estos cálculos no se pueden realizar de manera eficiente en un SGBDR, por lo que utilizar componentes especializados para esos cálculos es la mejor solución para la obtención de los resultados en un tiempo acorde con las necesidades de los usuarios.

condiciones de tráfico y los resultados de las rutas propuestas son almacenados por el sistema en bases de datos históricas que permiten monitorizar el tráfico y gestionarlo mejor en el futuro en función de los resultados obtenidos.

Desde el punto de vista tecnológico se utilizó la siguiente solución:

- Software:
  - **IBM InfoSphereTM Streams**, software encargado de la adquisición de los datos mediante flujo constante de estos desde los dispositivos generadores de datos hacia el sistema de captación y proceso de datos.
  - Sistema operativo **Red Hat Linux.**

- Hardware:
  - **Servidores** IBM BladeCenter HS22.
  - IBM System Storage® DS3400, solución externa de **almacenamiento.**

Según los informes disponibles, la implementación de *big data* en este proyecto supuso los siguientes beneficios en el sistema resultante:

- Estimación de tiempo de llegada a la destinación.
- Elección de las mejores rutas de tráfico en tiempo real.
- Mejora del tráfico en la zona metropolitana.

8.2. Incremento de la calidad de servicios de seguridad IT

Dell SecureWorks proporciona servicios de seguridad en el ámbito de las tecnologías de la información, permitiendo a sus clientes proteger sus entornos, cumplir con las regulaciones vigentes y reducir los costes en materia de seguridad. Estos servicios van desde accesos remotos a redes privadas, a protección de datos, pasando por el análisis de intentos de acceso y protección contra software malicioso, entre otros muchos.

En 2012, Dell SecureWorks ocupó un lugar de privilegio en su análisis de empresas proveedoras de servicios de seguridad (MSSP) en América del Norte.

Con el fin de poder mantener el nivel de servicio adecuado a sus clientes, 24 horas al día, 365 días al año, Dell SecureWorks debe obtener y analizar una gran cantidad de información de estos. Además, esta solución debe ser fácilmente escalable para poder crecer junto con el volumen de negocio de la empresa.

Con estos requerimientos funcionales debemos descartar una solución BI tradicional basada en procesos de ETL convencionales. La solución fue utilizar *big data*[9]. La justificación del uso de *big data* se basa en que:

- Los clientes generan un **gran volumen de datos**, que deben ser enviados a Dell SecureWorks para su tratamiento y análisis. Estos volúmenes provie-

(9)El caso de éxito completo puede descargarse en el siguiente enlace: http://www.cloudera.com/content/dam/cloudera/Resources/PDF/Dell_SecureWorks_Case_Study.pdf

nen de cualquier conexión de usuario a un sistema informático de cualquier cliente, cualquier intento de lectura o escritura sobre sus ficheros, consulta en alguna de sus bases de datos, etc. En un solo día, Dell SecureWorks puede llegar a procesar alrededor de 20.000 millones de eventos.

- Se necesita un **tiempo de reacción mínimo** a los eventos iniciales con el objetivo de minimizar las consecuencias de las amenazas en el ámbito de la seguridad en los clientes. Este análisis de los datos en un corto periodo de tiempo permite a Dell SecureWorks ofrecer soluciones a problemas de seguridad en un tiempo ínfimo. Si el análisis de los eventos tuviese lugar con un retraso de hasta 24 horas, el riesgo se incrementaría sustancialmente, pudiendo causar serios problemas en sus clientes.

- Se obtiene una **gran variedad de datos**, algunos de ellos **complejos o no estructurados**, como pueden ser ficheros de trazas de *firewalls*, de bases de datos, de servidores de autentificación, etc. Las manipulaciones que requieren estos tipos de datos para su análisis no pueden ser proporcionadas de manera eficiente por un SGBDR, ya que requieren de *parsers* de texto, búsqueda por palabras, y otras funcionalidades avanzadas. En cambio, un sistema *big data* puede utilizar rutinas de análisis de texto diseñadas especialmente para cada uno de los tipos de datos de entrada, lo que hace su análisis mucho más rápido y eficaz.

Desde el punto de vista tecnológico se utilizó la siguiente solución:

- Software:
  - Distribución de Apache Hadoop de Cloudera, como entorno de computación distribuida de *big data*.
  - Dell Crowbar software *framework*, un entorno para el despliegue de soluciones en entornos distribuidos.
- Hardware:
  - **Servidores** optimizados **Dell PowerEdge C**, con una capacidad por servidor de hasta 38 TB de disco y hasta 192 GB de memoria RAM.
  - Solución de **red Dell Force10**, para una interconectividad a altas velocidades dentro del clúster de Hadoop.

El resultado de implementar *big data* para satisfacer los requerimientos funcionales iniciales supuso la obtención de los siguientes beneficios, según el caso de estudio:

- Reducción del coste de almacenamiento en más de un 98%, de cerca de $17/GB (17 dólares) a aproximadamente ¢21/GB (21 centavos de dólar).

- Facilidad de escalabilidad en el futuro, gracias a la implementación de Apache Hadoop.

- Reducción del tiempo de implementación, al usar hardware no especializado y software *open source*.

- Capacidad de asegurar alta disponibilidad para servicios de seguridad críticos para los clientes. Esta alta disponibilidad viene dada por la implementación de Hadoop, lo que incluye la replicación de datos en múltiples nodos del clúster Hadoop.

- Flexibilidad para analizar datos estructurados, complejos y desestructurados.

Resumen

En este módulo se ha definido el concepto de sistema de BI basado en *big data*, y cómo *big data* nos permite el análisis de datos más allá de las capacidades de un sistema BI tradicional.

Hemos visto cómo algunos sistemas pueden generar enormes volúmenes de datos o datos muy heterogéneos, cómo el análisis de estos puede ser crítico y cómo la toma de decisiones puede depender de datos complejos y desestructurados. En esta situación, un sistema BI tradicional basado en una carga de datos con ejecución periódica y un SGBDR para el almacenamiento de los datos puede no satisfacer los requerimientos de análisis dentro de los parámetros mínimos del sistema. En esta situación es donde *big data* debe ser implementado.

Se ha introducido la tecnología para entender cómo funciona *big data* y las diferencias con un sistema BI tradicional. En este punto, no se ha pretendido dar una descripción detallada a nivel tecnológico, pero sí se ha proporcionado suficiente detalle como para ofrecer unos conocimientos básicos que faciliten la comprensión de la tecnología detrás de *big data*. También se han indicado diferentes fabricantes de software que trabajan en soluciones *big data*.

Durante este módulo también se han dado consejos para el cálculo de las métricas de evaluación TCO, *ROI* y *payback period*.

Por último, se han mostrado un par de casos de éxito donde se ha mostrado los beneficios que ha reportado aplicar una solución de *big data* y el porqué de dicha elección.