

# INTRODUCCIÓN AL BIG DATA



## ÍNDICE

<b>TU RETO EN ESTA UNIDAD.....</b>	<b>3</b>
<b>1. REVISITANDO LA ERA DEL DATO .....</b>	<b>5</b>
<b>2. CARACTERÍSTICAS DEL BIG DATA.....</b>	<b>7</b>
2.1. VOLUMEN .....	8
2.2. VELOCIDAD.....	9
2.3. VARIEDAD .....	9
2.4. VERACIDAD .....	10
2.5. VALOR.....	11
<b>3. MÁS SOBRE TIPOS DE DATOS.....</b>	<b>12</b>
3.1. ESTRUCTURADOS.....	12
3.2. NO ESTRUCTURADOS .....	13
3.3. SEMIESTRUCTURADOS .....	14
<b>4. SISTEMAS BIG DATA .....</b>	<b>15</b>
4.1. HISTORIA DE UNA PLATAFORMA DE DATOS .....	15
4.1.1. UN PEQUEÑO SERVIDOR.....	16
4.1.2. DE CÓMO SE HIZO GRANDE .....	16
4.1.3. REPARTIR ES BUENO (PERO NO SENCILLO).....	16
4.1.4. ¿MUERTOS DE ÉXITO? .....	17
4.2. BIG DATA AL RESCATE. ....	17
4.3. PROPIEDADES DE UN SISTEMA BIG DATA.....	18
4.3.1. ESCALABILIDAD.....	18
4.3.2. LATENCIA .....	20
4.3.3. TOLERANCIA A FALLOS .....	21

<b>5. EL CICLO BIG DATA.....</b>	<b>23</b>
5.1. ADQUISICIÓN .....	23
5.2. PREPARACIÓN Y PROCESADO .....	24
5.3. ANÁLISIS Y MODELIZACIÓN.....	25
5.4. ALMACENAMIENTO .....	26
5.5. PUBLICACIÓN .....	27
<b>6. PROCESOS POR LOTES VS TIEMPO REAL.....</b>	<b>28</b>
6.1. PROCESOS POR LOTES.....	28
6.2. PROCESOS EN TIEMPO REAL.....	29
<b>7. TECNOLOGÍAS BIG DATA: UN ANTICIPO.....</b>	<b>30</b>
7.1. HADOOP .....	30
7.2. SPARK.....	31
7.3. NOSQL .....	32
<b>¿QUÉ HAS APRENDIDO? .....</b>	<b>35</b>
<b>AUTOCOMPROBACIÓN .....</b>	<b>37</b>
<b>SOLUCIONARIO .....</b>	<b>41</b>
<b>BIBLIOGRAFÍA.....</b>	<b>43</b>

## TU RETO EN ESTA UNIDAD

---

Al comienzo de este curso, en la primera unidad, ya tratamos de darte unas primeras pinceladas del origen y definición del Big Data, y por qué este campo no deja de crecer.

En esta unidad vamos a repasar algunos de los conceptos que ya te presentamos, profundizando y añadiendo algo más de contexto, antes de explorar las tecnologías Big Data más importantes en las siguientes unidades.



# 1. REVISITANDO LA ERA DEL DATO

---

El ritmo al que se generan nuevos datos en nuestro mundo digital es desbordante.


No hablamos solamente de creación de nuevo contenido como noticias, fotos, vídeos...

Nosotros somos generadores de datos sin darnos cuenta, con gestos cotidianos. Cada vez que:

- hacemos una búsqueda en Google.
- al visitar una página web, leer un blog, dejar un comentario.
- al enviar un correo electrónico.
- cuando vemos un video en YouTube o una serie en Netflix.
- al comprar en Amazon.
- al interactuar en redes sociales, o enviar mensajes, o subir fotos.
- o simplemente al pagar con la tarjeta en una tienda.
- o al llamar por teléfono...

estamos generando datos. Si pensamos en los cientos o miles de millones de usuarios en el mundo, cada minuto se están generando millones y millones de registros sobre nuestra actividad, nuestros gustos, la gente que conocemos o cómo consumimos.

Pero esta avalancha de datos no se genera únicamente en Internet. El sector financiero, la industria bioquímica y farmacológica, o campos de investigación como la secuenciación del genoma, la ingeniería de materiales o la física de partículas no dejan de producir datos.



Ojo al dato

Según un informe de IBM para 2017, actualmente se está creando más de 1 Exabyte nuevo de información al día. Estamos hablando a llenar un millón de discos duros de 1TB o 200 millones de DVDs cada día. Se estima que el 90% de los datos que hoy existen en el mundo se han creado en los últimos dos años. Y gracias al ritmo de crecimiento exponencial, la cantidad de información que se genera es cada vez mayor.

Hay que añadir el impacto de la nueva explosión del Internet de las Cosas (IoT, por *Internet of Things*) y la integración de sensores en cualquier máquina u objeto imaginable, comunicándose unos con otros de forma transparente para las personas. Coches y hogares inteligentes, sensores en electrodomésticos, en la ropa... Seguro que conoces a personas, tú mismo, que utilizan relojes o pulseras con sensores para monitorizar las pulsaciones, los pasos, la velocidad en la bicicleta... Son cientos de millones de elementos generando más y más datos.

Y para terminar, el surgimiento de la llamada Industria 4.0. Fábricas y talleres donde todas las máquinas están comunicadas e intercambian información de su estado y su rendimiento, con sensores que controlan parámetros propios y de los productos que generan para asegurar la calidad y que todo funciona bien. Toda esa información suministrada al instante para poder monitorizarla y analizarla.

Se dice que vivimos una inundación de datos. Más bien estamos ante la madre de todos los diluvios.

La cuestión es qué hacer con todos estos datos, cómo gestionarlos y poder sacarles provecho. No se trata únicamente de cómo almacenarlos, si no de conseguir transformarlos en conocimiento. Y en un conocimiento *accionable*, es decir, un conocimiento útil y con alto valor, que nos permita tomar decisiones cuando lo necesitemos.



## 2. CARACTERÍSTICAS DEL BIG DATA

Dar una definición breve de Big Data no es algo simple. En una primera aproximación podríamos entender como Big Data aquellos conjuntos de datos cuyo tamaño y complejidad excede la capacidad de los sistemas de software y hardware tradicionales para capturarlos, almacenarlos y procesarlos de forma eficiente y en un tiempo razonable.

En 2001 Gartner ya planteaba que los sistemas de información se enfrentaban a nuevos desafíos derivados de la explosión de datos masivos, identificando tres cuestiones principales: el elevado volumen de datos a gestionar, la elevada velocidad a la que se generan y la amplia variedad de tipos de datos distintos con los que trabajar simultáneamente.

A esto se le conoce como el modelo de las 3 V's del Big Data. Podemos considerar que estamos ante Big Data cuando tenemos que gestionar datos que cumplen al menos alguna de estas características.

Más tarde se les han ido añadiendo otras V's adicionales, como la *veracidad* de los datos, la *volatilidad*, la *variabilidad* o el *valor* de los análisis y resultados.



### Resumiendo

Volumen, velocidad, variedad, veracidad, volatilidad, variabilidad, valor...

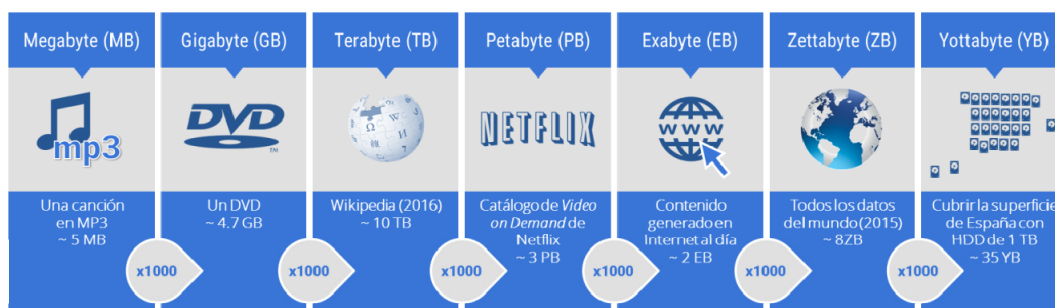
Si trabajas en Big Data, seguro que lo verás escrito en muchos sitios:

**Big Data se escribe con V**

Vamos a detallar un poco más qué implicaciones tienen algunas de las principales características.

## 2.1. VOLUMEN

La principal característica del Big Data es el tamaño o volumen de datos a manipular. Dependiendo del problema, podemos estar hablando de manejar desde decenas de Terabytes a varios Petabytes de datos nuevos de forma regular.



Ante este volumen de datos, el primer reto es cómo almacenarlos. Los sistemas tradicionales basados en almacenar la información en un servidor de ficheros o de base de datos no son viables en este caso. Incluso usando servidores con varios discos de gran tamaño, en poco tiempo llenaríamos todo el espacio disponible. La cantidad de datos crece mucho más rápido que la capacidad del mejor de los servidores.

La solución pasa por utilizar sistemas distribuidos, formados por varias máquinas o nodos interconectados por red, en los que la información se trocea y se replica para almacenarla y poderla recuperar después. Esta solución tiene también sus limitaciones, pero ofrece una virtud fundamental en el mundo *Big Data*, la escalabilidad. Si la cantidad de información crece, añadimos más nodos. Más adelante profundizaremos en este y otros puntos.

## 2.2. VELOCIDAD

Hay datos que se generan de forma periódica y estable, y que podemos almacenar y procesar con cierto margen de tiempo.

Pero también hay datos que se generan de manera continua, como el flujo de mensajes en una red social o de transacciones electrónicas en un gran comercio online. En estos casos en los que la actividad es continua y se crean nuevos eventos (datos) a muy alta velocidad, la dificultad ya no es sólo el almacenamiento. También el poder ejecutar procesos de manipulación y análisis de estos datos conforme se actualizan. Disponer de resultados basados en información en tiempo real es crítico en muchos negocios. Y eso implica disponer de tecnologías capaces de la ingesta y procesamiento de estos grandes flujos de datos de manera inmediata.

## 2.3. VARIEDAD

En un sistema de información normal es habitual que los datos no vengan de una única fuente. Puede haber una fuente principal, pero suelen incluirse datos complementarios de alguna otra fuente o sistema de información auxiliar. Los conocidos *Data Warehouses* no dejan de cumplir también con esa misión. Integrar datos de varias fuentes (de una empresa u organización) en un único *repositorio* y organizarlo de forma que toda la información sea fácil de acceder y consultar para un uso muy determinado.

La diferencia en Big Data es que no hablamos solamente de variedad de unas pocas fuentes, si no de variedad de formatos y estructuras de datos, provenientes de múltiples fuentes, cada una de ellas generando enormes cantidades de información.

Algunos datos vendrán de nuestras bases de datos de toda la vida, o de ficheros y hojas de cálculo. Pero otros datos pueden venir de teléfonos móviles, o de sensores, o de señales y *logs* de otras máquinas. Pueden ser registros con formatos y campos bien definidos, o ser textos de mensajes o páginas web, documentos, o elementos multimedia como fotos y vídeos.

Cuando tenemos que combinar y trabajar de forma conjunta con datos tan heterogéneos, las tecnologías convencionales (como las bases de datos relacionales o *data warehouses*) se encuentran con muchas dificultades, no soportan de forma sencilla y eficiente esta variedad de datos.

Necesitamos sistemas que tengan la capacidad para manejar múltiples tipos de datos de forma sencilla. Deben ser flexibles y poder adaptarse a nuevos datos sin que aumente la complejidad del sistema y sin afectar al rendimiento. Las tecnologías Big Data facilitan esta flexibilidad y la capacidad de integrar distintos tipos de información.

## 2.4. VERACIDAD

No todos los datos que tenemos que utilizar vienen *limpios*, listos y válidos para nuestros cálculos. Es normal que algunos datos presenten ciertos problemas, como valores inexactos, ausentes o completamente sin sentido.

Cuando estamos procesando unos cientos o miles de datos puede resultar sencillo detectar y corregir estos casos. Cuando tenemos que procesar millones y millones de registros de forma regular, lidiar con estos valores es crítico y complejo.

Los sistemas Big Data permiten abordar estas tareas de forma más sistemática para poder marcar y filtrar los datos problemáticos y tenerlo en cuenta de cara a los análisis posteriores.



## 2.5. VALOR

¿De qué sirve desarrollar un sistema Big Data para procesar y analizar cientos de millones de datos si los resultados finales no son aprovechables, si después de todo no podemos sacar rendimiento y utilidad al esfuerzo e inversión tan grandes?

Evaluar continuamente la calidad de los modelos y resultados para determinar su utilidad y valor es clave para poder actuar en consecuencia. Y ante avalanchas de datos, resulta complejo de medir.

## 3. MÁS SOBRE TIPOS DE DATOS

---

Como acabamos de ver, un aspecto característico del Big Data es su capacidad para integrar y manejar de forma conjunta datos de naturalezas muy diferentes.

Para analizar el éxito de sus productos y hacer recomendaciones a sus clientes, un comercio puede utilizar los datos de ventas, pero también las visitas a su web, el número de visualizaciones de cada producto y de sus fotos, los *clics* en *banners* y ofertas, las valoraciones y comentarios de opinión, los mensajes en redes sociales, etc.

Algunos datos son simples números, otros son textos, hay información que se extraerá de otras bases de datos, hay imágenes...

Para tratar de poner algún orden y manejar cada tipo de dato de la forma más adecuada, solemos clasificar estos datos en estructurados y no estructurados.

### 3.1. ESTRUCTURADOS

Los datos estructurados son aquellos que tienen un formato, un tamaño y una organización conocidas. Campos numéricos, fechas o cadenas de texto (para almacenar un nombre o una dirección) son datos estructurados. También un registro compuesto por distintos campos, o una colección de valores o registros. Piensa en una tabla de una base de datos o una hoja de cálculo.

Los datos estructurados suelen ser los más comunes en los sistemas de información convencionales de empresas, en *data warehouses* o sistemas de gestión de clientes (*CRMs*) y procesos (*ERPs*).

Pero también podemos tener datos estructurados de otras fuentes, como formularios web, *logs* de actividad de servidores, o datos enviados por sensores.

El medio más habitual y conocido para garantizar la persistencia de estos datos y facilitar su acceso y consulta es el uso de bases de datos relacionales. Como ya sabes, en estas bases de datos la información se organiza y almacena en tablas. Cada tabla representa un tipo de dato o entidad (p.ej. una tabla para productos, otra para clientes), y está compuesta por varias columnas o campos, que representan distintos atributos de una entidad. Además, se pueden definir relaciones entre tablas a través de algún campo común que las vincule.

Los sistemas de bases de datos relacionales siguen siendo la mejor solución para una gran parte de las necesidades de gestión de datos de muchas aplicaciones.

Sin embargo, cuando saltamos a la *escala Big Data*, estos sistemas tienen muchas dificultades para adaptarse y rendir bien ante la avalancha de datos.

Además, los datos deben ajustarse a un modelo o esquema tabular. Pero hay datos estructurados que no encajan en este patrón, por ejemplo datos jerárquicos (tipo árbol) o el grafo de una red. Si queremos incorporar estos datos debemos “inventar” representaciones alternativas que sí encajen en el modelo de tablas, normalmente aumentando la complejidad y perdiendo eficiencia en los cálculos.

## 3.2. NO ESTRUCTURADOS

Los datos no estructurados son aquellos que no siguen un modelo o esquema predefinido, ni un tamaño delimitado, ni sigue ninguna organización lógica.

El ejemplo de dato no estructurado más sencillo y común es el contenido de texto. El contenido de una página web, los comentarios de un foro, el cuerpo de un correo electrónico, un mensaje instantáneo...

También son datos no estructurados los contenidos multimedia: imágenes, vídeos, audio.

A día de hoy, todo este contenido no estructurado supone más del 80% de la información total que existe en el mundo.

Las tecnologías Big Data se aplican en estos casos para tareas como reconocimiento del lenguaje, traducción automática, análisis de sentimiento, procesado y reconocimiento de imágenes, etc.

### 3.3. SEMIESTRUCTURADOS

En ocasiones se distingue una categoría adicional, los datos semiestructurados. Podríamos definirlos como aquellos datos que no tienen una estructura fija inmutable, como los datos tabulares, pero que sí siguen cierto esquema o estructura flexible, con elementos internos diferenciados.

Los datos semiestructurados permiten representar información con un patrón jerárquico o anidado de elementos, o incluir campos formados por listas de valores arbitrariamente largas, o incluso modelos más complejos, como grafos.

En realidad, se tratan de datos muy extendidos en el ámbito de las aplicaciones web y la comunicación entre distintos servicios en red. Los dos principales formatos para representar esta información son XML y JSON.

Muchas de las tecnologías Big Data existentes son capaces de trabajar con datos en estos formatos de forma directa, simplificando el manejo y acceso a la información que contienen.





## 4. SISTEMAS BIG DATA

---

Por extensión, también solemos referirnos como Big Data a las distintas tecnologías diseñadas especialmente para poder trabajar con datos de esta naturaleza.

¿Qué distingue a los sistemas Big Data de un sistema de información convencional?

### 4.1. HISTORIA DE UNA PLATAFORMA DE DATOS



**Ejemplo**

Para ponernos en situación y explicarte mejor qué hace diferente a un sistema Big Data, vamos a plantear primero un caso de ejemplo.

Imagina que tienes que diseñar un sistema para almacenar y manejar la información para tu nueva tienda *online*. Por simplificar, ignoraremos toda la parte de la aplicación web y nos vamos a centrar en cómo manejar la información. Tienes datos de tus productos, de los clientes y los pedidos. El sistema tiene que poder almacenarlos de forma organizada y eficiente, y que nos permita buscar y operar después con los datos de manera fácil y rápida.

### 4.1.1. UN PEQUEÑO SERVIDOR

La opción natural para empezar es montar un pequeño servidor con una base de datos sencilla. Organizamos nuestros datos en tablas y montamos los procesos y consultas sobre la base de datos. Esta solución cubre todas nuestras necesidades iniciales.

### 4.1.2. DE CÓMO SE HIZO GRANDE

Tu tienda *online* empieza a tener éxito, el número de visitas y clientes crece. Aumentas tu oferta de productos para satisfacerlos, se incrementan los pedidos. Y los registros empiezan a llenar tu base de datos. La solución *tradicional* es ampliar tu servidor o directamente cambiarlo por uno superior. Más memoria, más disco, mejor procesador. Y más caro, claro.

### 4.1.3. REPARTIR ES BUENO (PERO NO SENCILLO)

Sigues creciendo, y tu gran servidor también se queda pequeño. No es posible ampliarlo más, ni resulta rentable comprar un superordenador. ¿Cuál es el siguiente paso?

Llegados aquí, la estrategia a aplicar suele ser la de *divide y vencerás*. Es decir, partir la información, dividir los datos y repartirlos entre varios servidores similares al que tienes, interconectados entre sí. Lo que llamamos un *cluster* de máquinas. La idea es que la carga del almacenamiento y procesamiento de los datos se comparta de forma más o menos equitativa entre los servidores del *cluster*. En este punto pasamos a utilizar bases de datos distribuidas, en lugar de nuestra base de datos individual. A este tipo de partición de datos se la denomina también *sharding*.

Sin embargo, gestionar el sistema ahora empieza a ser mucho más complejo. Hay que decidir con qué criterio se envía un dato a un servidor o a otro, redistribuir los datos cada cierto tiempo para asegurarse que la carga es similar, o vigilar que no caiga uno de los servidores, porque perderíamos parte de la información. Y además, tampoco es trivial ampliar el sistema *en caliente*, mientras todo está funcionando.

#### 4.1.4. ¿MUERTOS DE ÉXITO?

Ahora imagínate que el boca a boca en las redes sociales hace que el éxito de tu tienda *online* sea literalmente *desbordante*. Los clientes y pedidos, es decir, el tráfico de datos a almacenar y procesar crecen a un ritmo exponencial.

Ya has visto las dificultades de nuestra base de datos particionada. Cada vez que tenemos que añadir un nuevo servidor al cluster hay que volver a repartir los datos, asegurarnos de que la carga está equilibrada, comprobar la corrección final del sistema. Ni por coste ni por complejidad podemos ampliar indefinidamente un sistema de este tipo. ¿Qué hacemos?

### 4.2. BIG DATA AL RESCATE.

El enfoque en las tecnologías Big Data no deja de basarse esencialmente en la misma idea de distribuir el trabajo, en la línea del reparto de datos y carga que hemos comentado con las bases de datos distribuidas.



**Importante**

Tanembaum definió un sistema distribuido como una colección de ordenadores interconectados que aparentan ser un sistema único y coherente para el usuario.

Pero entonces, ¿qué hace diferente al Big Data?

La idea es que la existencia de varias máquinas, cada una corriendo con parte de los datos, y que se comuniquen entre ellas a través de una red para completar los cálculos, debería ser algo transparente para el usuario final. Son detalles que no tendrían que afectarle para realizar su trabajo.

La realidad es que en sistemas como las bases de datos particionadas que hemos mencionado (y otras soluciones similares), esa distribución de datos y de ejecución de procesos no es completamente transparente. En la mayoría de los casos, sólo lo es de forma parcial.

El usuario debe ser consciente al diseñar su modelo de datos y los procesos de cálculo de cuál es la configuración del sistema, saber que existen varios nodos; pensar y configurar cómo repartir datos y procesos para asegurarse de que la carga esté equilibrada, etc.

Cuando tenemos que gestionar millones y millones de datos que no dejan de llegar, queremos poder centrarnos en las actividades que dan valor: procesar los datos, analizar resultados y desarrollar nuevas funcionalidades. No queremos ser engullidos por todas las complejidades de la gestión de un sistema distribuido.

Los sistemas Big Data son distribuidos de forma nativa, están concebidos y diseñados así desde el principio. No tienes que preocuparte de cómo hay que dividir los datos, ni de redistribuir o replicar la información, ni de complejos métodos para añadir nuevas máquinas o desligar otras defectuosas. Las plataformas Big Data se encargan de la mayoría de estos procesos de forma autónoma, casi invisible, con poca o mínima intervención del usuario. Así nos facilitan enormemente la gestión y operación normal del sistema.

## 4.3. PROPIEDADES DE UN SISTEMA BIG DATA

### 4.3.1. ESCALABILIDAD

Aparte de poder procesar enormes cantidades de datos, un sistema Big Data debe poder crecer para adaptarse a mayores cargas de información, manteniendo su rendimiento y capacidad de respuesta sin degradar.

A esta capacidad de crecer y adaptarse a cargas de trabajo mayores la denominamos *escalabilidad* del sistema.

Podemos escalar un sistema añadiendo más recursos a una máquina (lo que comúnmente llamamos *ampliación*). Podemos aumentar la memoria RAM, añadir más discos duros, o directamente reemplazar la máquina por una superior. Esto se conoce como *escalar verticalmente*; hacemos más grande una máquina particular.

Si estamos en un sistema distribuido formado por varios nodos, podemos incrementar la capacidad de almacenamiento y cálculo global del sistema añadiendo nuevas máquinas al conjunto. Esto se conoce como *escalar horizontalmente*.



**Escalado Vertical**



**Escalado Horizontal**

Como ya hemos dicho, añadir nuevos nodos a un sistema distribuido tradicionalmente no ha sido una tarea simple, y podía requerir bastante trabajo de configuración y puesta a punto del sistema para integrar el nuevo servidor. Además del coste de la máquina había que tener en cuenta el coste de este trabajo. Había que planificar muy bien el crecimiento.

Sin embargo una de las ideas motivadoras detrás de las actuales tecnologías Big Data fue poder construir sistemas con una gran potencia de cálculo, aprovechando un gran número de servidores “baratos” interconectados, en lugar de usar unos pocos servidores de altas prestaciones. Ampliar el sistema debía ser cuestión de añadir unos cuantos equipos “asequibles” más siempre que hiciera falta. Y estas ampliaciones debían de ser fáciles y rápidas de llevar a cabo.

Los sistemas Big Data están diseñados para añadir nuevos nodos de forma mucho más sencilla. De hecho, muchas plataformas tienen capacidad de descubrimiento automático de nodos y reconfiguración de la red de servidores, integrando las nuevas máquinas.

Básicamente, puede bastar con instalar el software en el servidor, configurarlo y conectarlo a la misma red, *et voilà*, en unos segundos la plataforma se habrá encargado de incluirlo. Los procesos de réplica y redistribución de los datos también se pondrán en marcha de forma automática, sin ninguna intervención adicional del usuario.

Es más, en algunos casos estos sistemas también son capaces de reconocer si el nuevo servidor tiene más o menos capacidad, y repartir la carga de datos y procesos de forma más ajustada.

### 4.3.2. LATENCIA

La latencia es la suma de retardos o tiempos consumidos por distintas partes del sistema para completar una tarea. Por ejemplo, tenemos una latencia de acceso a disco para leer los datos, una latencia de red al transmitirlos, etc.

Cuando trabajamos con grandes volúmenes de datos, estos tiempos empiezan a crecer y son una cuestión clave. Imagina lo que puede suponer si para nuestros análisis tenemos que leer de disco gigas o terabytes de datos y transferirlos de una máquina a otra, una vez y otra vez.

Muchas aplicaciones necesitan que la latencia sea baja, de forma que el tiempo de respuesta al usuario se mantenga aceptable. Para otros cálculos que solo se necesiten periódicamente, una latencia superior (p.ej. de horas) puede ser perfectamente válida.

Distribuir la carga de datos y trabajo permite ajustar mejor las latencias totales del sistema. Aquí entra otra parte clave de las tecnologías Big Data. Procurar que los cálculos se ejecuten donde están los datos, a ser posible en las mismas máquinas. De poco nos sirve tener los datos repartidos entre múltiples máquinas si después hace falta transferirlos a otros nodos para procesarlos.

Dentro del mundo Big Data existen distintas tecnologías adaptadas para diferentes casos. Tenemos tecnologías indicadas para ejecutar pesados procesos de datos por lotes, sistemas de trocean en pequeñas unidades los datos, o sistemas que operan de forma rápida sobre los datos individuales conforme llegan. Dependiendo de las características de nuestro problema, podremos combinar varias tecnologías para cubrir nuestras necesidades.

La circunstancia más extrema es cuando nos encontramos con flujos de datos que llegan de forma continuada y a gran velocidad (como eventos o señales de sensores), y nuestra aplicación debe procesarlos y responder de manera inmediata, en tiempo real. La latencia de nuestro sistema debe ser mínima. Tenemos tecnologías Big Data especializadas en *stream processing* y *event processing*.

### 4.3.3. TOLERANCIA A FALLOS

La naturaleza distribuida de los sistemas Big Data los hace complejos. Tenemos muchos nodos repartidos los datos y tareas de cálculo, y coordinándose entre sí. Hay muchos puntos donde puede producirse un fallo.

Imagina que una de las máquinas falla, o cae su conexión a la red. Perderíamos el acceso a los datos que almacene y también perderíamos su capacidad de proceso, no podrá colaborar en nuevos cálculos. Incluso una pequeña interrupción momentánea puede provocar que se pierda un dato o un resultado del cálculo de un nodo.

Nos gustaría que el sistema fuera *immune* a estos fallos, que sea capaz de detectar y sobreponerse a problemas de este tipo, y siga funcionando correctamente, completando las tareas sin errores desde el punto de vista del usuario.

Pero conseguir esto es complejo: implica cuestiones como crear réplicas de los datos en varios nodos para que haya más copias disponibles, controlar y coordinar las transferencias de datos y los cálculos para evitar pérdidas, detectar incoherencias entre nodos, evitar solapes o duplicidades en los resultados, etc. Además, hay que monitorizar la buena salud de los nodos y, si eventualmente se detecta un problema con uno de ellos, replanificar y distribuir la carga de datos y de cálculo entre el resto.

Una de las claves en los sistemas Big Data consiste en incorporar los mecanismos necesarios para protegerse y gestionar todos estos aspectos de forma automática. El objetivo es doble: minimizar el tiempo que el sistema deje de estar operativo por una falla, y descargar todo lo posible a los responsables del sistema de las complejidades de mantenerlo en funcionamiento de forma correcta.



### Resumiendo

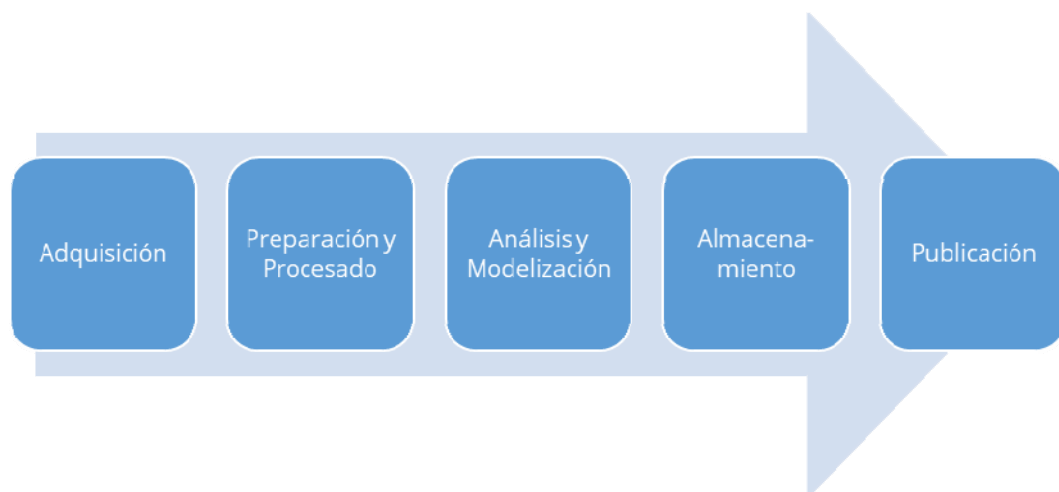
- **Escalabilidad:** el sistema debe ser capaz de ampliarse y crecer de forma sencilla para adaptarse a un incremento en la carga de trabajo, manteniendo su rendimiento.
- **Latencia:** los tiempos de respuesta del sistema deben mantenerse ajustados, según los requerimientos. El sistema debe estar diseñado para que la latencia sea adecuada ante grandes volúmenes de datos, o con flujos de entrada a gran velocidad.
- **Tolerancia a fallos:** El sistema debe continuar funcionando correctamente ante fallos o indisponibilidad de alguno de sus nodos. Los mecanismos para asegurar el funcionamiento y reponerse ante un fallo deben ser transparentes para el usuario.



## 5. EL CICLO BIG DATA

---

Un sistema Big Data debe proveer los medios para facilitar ciertas tareas comunes. A continuación te enumeramos algunos de los principales pasos de un flujo de datos típico en un sistema Big Data. Si buscas, verás que dependiendo de la fuente consultada, incluyen más o menos pasos.



### 5.1. ADQUISICIÓN

Es el proceso de captura de los datos, incluyendo algún preproceso básico de filtrado y limpieza, antes de incorporarlos en el sistema. La adquisición de datos puede ser uno de los puntos más críticos. Es necesario que la infraestructura permita la llegada e ingesta de grandes volúmenes de datos manteniendo una baja latencia, para evitar colapsos y pérdida de información.

Además, debe tener flexibilidad para aceptar datos con estructuras diversas sin que ello afecte a su desempeño.

## 5.2. PREPARACIÓN Y PROCESADO

Antes de poder realizar labores de análisis, es necesario preparar los datos para que su uso y explotación posterior sea lo más ágil posible.

Los datos capturados pueden contener valores incorrectos o ausentes. Puede haber diferencias en la codificación de algunos campos al trabajar con múltiples fuentes (p.ej. unos datos pueden contener velocidades de viento en m/s y otros en km/h; unos datos pueden contener el nombre de un municipio y otros un código postal). También es común que entre los grandes volúmenes de información exista información duplicada.

Hay que realizar tareas comunes de limpieza, validación, normalización, filtrado. También precalcular distintas agregaciones que podamos necesitar después. En general, cualquier transformación básica de los datos que nos permita tener los datos listos para su análisis.

Estos procesos no son distintos de los que harías al trabajar con datos en cualquier sistema de información convencional. La complejidad, como siempre, es la cantidad de datos a manejar. Completar todas estas tareas sobre millones y millones de registros puede consumir mucho tiempo y recursos.

Piensa que cuando vayas a analizar o utilizar los datos más adelante en tus aplicaciones, querrás obtener respuestas lo más rápidamente posible. Pero si tienes que hacer cualquiera de estas transformaciones durante el análisis, el tiempo empezará a acumularse.

Por eso hay que planificar los principales preprocesos y adelantar su ejecución, almacenando los datos ya preparados. De hecho, es una práctica común guardar múltiples versiones de los mismos datos, con distintas transformaciones y agregaciones.

El espacio de disco es barato. El tiempo (de CPU, de respuesta al usuario) es caro. Es mejor tener información replicada que recalcular.

En un sistema Big Data, estas tareas suelen realizarse mediante procesos por lotes que se ejecutan de forma regular sobre un gran conjunto de datos. Estos datos suelen estar organizados en grandes archivos, almacenados en un sistema de ficheros distribuido. La tecnología más extendida para dar solución a estas dos funcionalidades es Hadoop.

### 5.3. ANÁLISIS Y MODELIZACIÓN

Esta es la parte donde se crea el principal valor añadido de tu aplicación.

Implica explorar, descubrir patrones y relaciones entre los datos, sintetizar y crear los modelos para nuestras aplicaciones, ya sean de clasificación, predicción, recomendación, etc.

De nuevo, repasando estas tareas, no verás nada distinto de un proyecto de ciencia de datos normal.

Pero a estas alturas tú ya sabes dónde está el problema diferencial. No es lo mismo explorar la distribución de cien valores que de cien millones. No es lo mismo analizar la relación entre dos variables con mil observaciones y ajustar un modelo, que hacerlo con cien variables con un millón de observaciones cada una.

Estas tareas requieren de cálculos complejos, suele ser necesario hacer múltiples pruebas, simulaciones y validaciones cruzadas sobre los mismos datos antes de llegar al modelo final. Se trata de procesos que requieren interacción del científico de datos, se basan muchas veces en el ensayo y error.

Cuando estamos explorando datos, relaciones y modelos, necesitamos agilizar los cálculos en la medida de lo posible. Que los tiempos de respuesta sean razonables para poder trabajar de forma más o menos interactiva.

Pero no queremos enfrentarnos a complejidades técnicas sobre cómo ajustar un modelo estadístico cuando los datos están distribuidos en varios nodos. Queremos calcular estadísticas o hacer una regresión igual que lo haríamos con unos pocos datos en nuestro ordenador.

Es necesario disponer de un sistema de computación distribuida que permita realizar y repetir cálculos complejos, trabajando de forma iterativa sobre los datos, con eficiencia y tiempos de respuesta acotados. Conviene que este componente permita una programación interactiva del usuario. Hoy en día, la plataforma Big Data de referencia para estas tareas es Spark.

## 5.4. ALMACENAMIENTO

Necesitamos almacenar los datos de origen que capturamos, los datos procesados y los resultados que generamos con nuestros análisis y modelos.

Necesitamos garantizar la persistencia de toda esta información, ya que de ello depende poder ejecutar nuestros procesos y análisis. Pero también para poder publicar y hacer accesibles nuestros resultados, y para replicar y auditar cada paso, cada operación.

Ya hemos comentado la opción de almacenar los datos de entrada en un sistema de ficheros distribuido. Sin embargo, recuerda que los datos de entrada pueden venir en una gran variedad de formatos. A lo que hay que añadir los datos procesados que generemos, más los datos resultados de nuestros modelos, cada cual con una estructura propia.

El repositorio de datos que utilicemos debe cubrir algunas características que ya conocemos:

- Debe ser flexible para poder integrar y trabajar con distintos tipos de datos.
- Debe ser capaz de gestionar grandes volúmenes de información de manera eficiente.
- Debe ser escalable, para adaptarse a crecimientos en los datos a manejar.
- Debe permitir acceder a la información con unos tiempos de respuesta ajustados.
- Debe facilitar un mecanismo para consultar y filtrar de forma sencilla los datos almacenados.
- Debe garantizar la persistencia, disponibilidad y recuperación de los datos.

Las bases de datos NoSQL han sido diseñadas teniendo todos estos objetivos en cuenta, muy especialmente la escalabilidad. Existen multitud de opciones, algunas con gran flexibilidad para almacenar distintos tipos de datos, y otras soluciones de alto rendimiento especializadas en modelos de datos concretos.

De hecho, en soluciones de cierta envergadura es habitual utilizar varias tecnologías NoSQL para cubrir necesidades distintas del sistema.

## 5.5. PUBLICACIÓN

Por último, los datos y resultados de nuestros análisis y modelos deben ser visibles y aprovechables, bien dentro de nuestra aplicación, como para otros usuarios.

Si hemos generado un modelo de previsión o un modelo de recomendación, deberá ser accesible para integrar su funcionamiento con las aplicaciones que lo necesiten. Una solución típica consiste en publicar sus resultados mediante servicios web.

Pero para aprovechar el conocimiento y poder tomar nuevas decisiones, también necesitaremos poder visualizar los datos de forma interactiva, crear informes o cuadros de mando gráficos.

De nuevo, las herramientas tradicionales no funcionan correctamente con la cantidad de datos que gestionamos en un sistema Big Data. De hecho, la mayoría no están adaptados para trabajar con sistemas NoSQL.

No obstante, existen numerosas tecnologías especializadas en la visualización y exploración gráfica para Big Data, así como para la publicación online de informes, o para crear y compartir cuadros de mandos interactivos.

## 6. PROCESOS POR LOTES VS TIEMPO REAL

---

Como no hemos dejado de explicarte, uno de los grandes beneficios que brindan las tecnologías Big Data es poder analizar cantidades de datos gigantescas de forma sencilla para poder tomar decisiones y emprender nuevas acciones en respuesta.

De hecho, la capacidad de poder procesar semejantes volúmenes de datos en tiempo real ha revolucionado la forma de analizar y responder a distintos problemas. Innumerables aplicaciones en todo tipo de campos se benefician de esta potencia.

Sin embargo, no todo es procesado en tiempo real. Cuando hablamos de la latencia de los sistemas Big Data ya te adelantamos que es posible distinguir procesos en función del tiempo de respuesta.

En general, podemos distinguir entre dos extremos: los procesos por lotes y los procesos en tiempo real.

### 6.1. PROCESOS POR LOTES

Los procesos por lotes (*batch processing*) o procesos planificados son aquellos en los que acumulamos datos y periódicamente ejecutamos los cálculos sobre el volumen completo de nuevos datos. Cuando el proceso concluye, los nuevos resultados son registrados para que el sistema pueda explotarlos y publicarlos.

Este método de procesamiento se aplica fundamentalmente cuando tenemos que tratar con grandes volúmenes de datos, pero no necesitamos disponer de nuevos resultados de forma inmediata.

Por ejemplo, si queremos calcular cuáles han sido las páginas o productos más vistos de nuestra web, o si queremos analizar cuáles han sido los términos de búsqueda más usados, no necesitamos calcularlo constantemente ni conocer los últimos cambios inmediatamente. Podemos ejecutar nuestros procesos periódicamente y seguir su evolución.

## 6.2. PROCESOS EN TIEMPO REAL

También llamados procesamiento continuo o *stream processing*. Hablamos de *tiempo real* cuando necesitamos que los nuevos datos se procesen inmediatamente conforme llegan al sistema, actualizando los resultados de forma continua. El tiempo de respuesta debe ser mínimo.

El flujo (*stream*) de datos de entrada en estos casos suele ser ininterrumpido, lo que obliga a ese procesamiento prácticamente en tiempo real para evitar que se acumulen retrasos.

Este modelo se utiliza en aplicaciones donde la respuesta debe ser inmediata. Por ejemplo, para detectar el uso fraudulento de tarjetas de crédito, o para recomendar actividades o lugares en base a la localización.

## 7. TECNOLOGÍAS BIG DATA: UN ANTICIPO

---

En las próximas unidades profundizaremos un poco más en algunas de las tecnologías Big Data más importantes.

### 7.1. HADOOP

Apache Hadoop es una plataforma software para trabajar con grandes volúmenes de datos de forma distribuida. Está especialmente diseñada para trabajar con *clusters* formados por un gran número de servidores comunes, facilitando la escalabilidad de la plataforma y garantizando la tolerancia a fallos en nodos.

Aunque está formado por varios componentes, los dos elementos principales de Hadoop son su sistema de ficheros distribuido (HDFS) y su motor MapReduce.

HDFS (*Hadoop Distributed File System*) es un sistema de ficheros distribuido preparado para almacenar archivos de gran tamaño (desde varios Gigabytes hasta Terabytes) a lo largo de múltiples máquinas. Los ficheros se trocean y reparten en bloques entre los nodos del *cluster*. Para garantizar la fiabilidad del sistema, HDFS almacena múltiples réplicas de cada bloque en nodos distintos.

Hadoop MapReduce es un motor para ejecutar trabajos de forma distribuida sobre el *cluster* de nodos de cálculo. Los trabajos deben seguir el modelo de programación *Map-Reduce*. En este modelo, una tarea debe dividirse esencialmente en dos pasos. En el primer paso (*map*) los datos se trocean y se realiza algún cálculo (transformación) individual sobre cada dato. En el segundo paso (*reduce*), se realiza una operación de agregación o combinación de los resultados, agrupados por una clave o variable.



Un gestor de tareas se encarga de planificar los trabajos y lanzarlos sobre los nodos disponibles. Los datos de entrada habitualmente se leen de ficheros almacenados en HDFS, por lo que es crítico repartir los cálculos entre los nodos que almacenan los bloques de datos a procesar. Cada nodo de cálculo ejecuta el proceso *map* sobre su porción de datos locales. Después, los resultados intermedios se reparten entre los nodos, agrupados por una clave o variable, y se aplica el proceso *reduce* sobre cada grupo. Estos valores finales se devuelven y combinan en un único conjunto de resultados.

El modelo de Hadoop es especialmente eficaz para implementar procesos de cálculo y agregación sobre grandes cantidades de datos, donde no importa que el tiempo de respuesta sea elevado. Es una solución perfecta para el tipo de procesamiento por lotes.

## 7.2. SPARK

Apache Spark es una plataforma de computación distribuida optimizada para ejecutar algoritmos complejos sobre grandes volúmenes de datos de forma sencilla y eficiente, aprovechando un *cluster* de máquinas para paralelizar datos y cálculos, garantizando la tolerancia a fallos en nodos del sistema.

Spark se caracteriza por trocear y distribuir los datos entre los nodos, utilizando unas estructuras de datos en memoria para manejarlos, llamadas *RDDs* (*Resilient Distributed Dataset*). Se trata de bloques o conjuntos de datos *inmutables* (no se modifican una vez creados), distribuidos y replicados entre los nodos del *cluster*, y mantenidos en memoria para optimizar el rendimiento en los cálculos. Eventualmente, al completar un cálculo, se puede forzar la persistencia de los datos en un almacenamiento permanente (sistema de ficheros o base de datos).

Como los RDDs son inmutables, cada vez que se completa una operación y se produce un resultado intermedio, se genera un nuevo conjunto de RDDs para representar los nuevos datos. De esta manera, se mantiene la historia y traza de todos los pasos.

Trabajar con grandes bloques de datos en memoria no solo reduce de forma drástica el tiempo necesario para ejecutar un cálculo individual. Además, mantener la traza de RDDs inmutables permite crear flujos de datos complejos manteniendo un elevado rendimiento.

El modelo de cálculo de Spark facilita dos tareas críticas en la ciencia de datos:

- El análisis exploratorio interactivo de los datos.
- La implementación de algoritmos iterativos sobre los mismos conjuntos de datos. Estos procesos de repetición de cálculos sobre un mismo conjunto de datos variando ciertos parámetros son típicos en el entrenamiento y validación de modelos estadísticos.

Para simplificar las cosas, Spark ofrece también la posibilidad de utilizar una capa de abstracción para trabajar con los datos bajo la forma de DataFrames. También incluye librerías implementando muchas de las principales técnicas y algoritmos de análisis estadístico y *machine learning*.

Spark además permite la posibilidad de trabajar con una sesión interactiva mediante un intérprete o consola, de forma similar a las consolas de R o Python. Así podemos ir probando nuestro código paso a paso.

## 7.3. NOSQL

El término NoSQL (o *Not Only SQL*) en realidad abarca a un sinnúmero de sistemas de bases de datos muy variados que tienen como característica común que no siguen el modelo de base de datos relacional de forma estricta.

Es decir, su modelo de datos no se basa en una estructura de tablas bien definidas, con relaciones entre ellas. Y por tanto, tampoco siguen el estándar SQL para acceder y consultar sus datos.

Aunque han existido algunas bases de datos NoSQL desde hace décadas, el *boom* del Big Data ha provocado su crecimiento y mejora hasta hacerlas casi ubicuas. Ejemplos de algunas de las bases de datos NoSQL más populares son MongoDB, Cassandra, Redis, HBase o Neo4J. Cada una con propiedades distintivas.

Los principales motivos de su éxito y expansión son varios, todos relacionados con los problemas del Big Data que ya hemos discutido en esta unidad:

- La necesidad de manejar datos muy diversos que no encajan en el modelo relacional.
- La capacidad para gestionar enormes volúmenes de datos y escalar de manera sencilla el sistema.
- Asegurar el acceso y disponibilidad de la información, que cada petición sea respondida en un tiempo adecuado.

El primer punto se resuelve en los sistemas NoSQL como ya te hemos contado, implementando modelos de datos no tabulares y más flexibles. En la unidad sobre NoSQL veremos los diferentes tipos de esquemas o modelos de datos que existen.

Los dos puntos restantes hacen referencia a la escalabilidad, la latencia y tolerancia a fallos del sistema. Las bases de datos NoSQL son sistemas distribuidos, están diseñadas desde cero para trabajar de forma distribuida y facilitar la escalabilidad, la replicación y garantizar la respuesta del sistema. Todo ello con mínima intervención del administrador y sin necesidad de ningún conocimiento extra por parte del usuario.



## ¿QUÉ HAS APRENDIDO?

---

Ahora que has terminado esta unidad ya conoces mejor los principales aspectos que diferencian a los sistemas Big Data.

Has aprendido cuáles son las características que nos permiten distinguir que estamos ante un problema de Big Data: un volumen elevado de datos, elevada velocidad de generación, gran variedad de formatos...

También tienes claro cómo reconocer los distintos tipos de datos: estructurados, no estructurados, semiestructurados.

Has aprendido cuáles son las principales propiedades de un sistema distribuido de tipo Big Data: la escalabilidad, la baja latencia y la tolerancia a fallos. Y conoces las diferencias e implicaciones que hay entre los procesos por lotes y de tiempo real.

Has visto las etapas típicas de un flujo de trabajo Big Data, y qué tipo de tecnologías o soluciones son más adecuadas en cada caso.

Por último, te hemos presentado brevemente las principales tecnologías Big Data, en las que profundizaremos en las siguientes unidades.

Esperamos que esta introducción al Big Data te haya resultado muy interesante y que haya aumentado tus ganas de aprender más sobre el tema. ¡Ánimo, que todavía queda lo mejor!



## AUTOCOMPROBACIÓN

---

### 1. ¿Cómo describirías qué es el Big Data?

- a) Son los conjuntos de datos que no caben en la memoria RAM de un servidor.
- b) Son los conjuntos de datos que no caben en el disco duro de un servidor.
- c) Son conjuntos de datos tan grandes y tan complejos que no pueden almacenarse y procesarse de forma eficiente con las aplicaciones y máquinas habituales
- d) Son conjuntos de datos tan grandes y tan complejos que no pueden almacenarse o procesarse con una base de datos tradicional o con una hoja de cálculo.

### 2. En Big Data el problema con el volumen de datos es:

- a) Que las necesidades de almacenamiento exceden las capacidades de una sola máquina.
- b) Que se necesitan varios discos duros en una máquina.
- c) Que no existen bases de datos adecuadas para manejarlo.
- d) Que se necesita una máquina con gran cantidad de RAM.

**3. Los datos no estructurados:**

- a) No siguen un modelo o esquema predefinido, y suponen más del 80% de la información existente en el mundo.
- b) Son el contenido multimedia, como videos o imágenes, y suponen más del 80% de la información existente en el mundo.
- c) Son los datos de redes sociales y mensajería instantánea, y suponen más del 80% de la información existente en el mundo.
- d) Son aquellos que no encajan en una base de datos relacional.

**4. ¿Cuáles son las principales propiedades de un sistema Big Data que hemos visto?**

- a) Escalabilidad, latencia y tolerancia a fallos.
- b) Escalabilidad, elasticidad y tolerancia a fallos.
- c) Elasticidad, velocidad y robustez.
- d) Escalabilidad, velocidad y robustez.

**5. Los sistemas Big Data están pensados para facilitar la escalabilidad:**

- a) Ampliando los recursos hardware de cada servidor.
- b) Añadiendo nuevos servidores económicos al sistema conforme sea necesario.
- c) Utilizando unos pocos servidores de altas prestaciones.
- d) Sustituyendo servidores antiguos por otros más potentes.

**6. ¿Cuáles son las etapas fundamentales de un flujo de datos típico en Big Data?**

- a) Adquisición, particionado, transformación, almacenamiento, análisis y publicación.
- b) Adquisición, particionado, replicación, procesado y recolección.
- c) Particionado, replicación, procesado, recolección y publicación.
- d) Adquisición, preparación y procesado, análisis y modelización, almacenamiento y publicación.



7. ¿Cuál de las siguientes *no* es una de las características generales a cubrir por un sistema de almacenamiento Big Data?
- a) Disponer de un mecanismo para consultar y filtrar los datos almacenados.
  - b) Ser escalable para adaptarse a crecimientos en los datos a manejar.
  - c) Garantizar la persistencia de los datos.
  - d) Compatibilidad con *data warehouses* y bases de datos relacionales previas.
8. ¿Cuándo utilizamos procesos por lotes en Big Data?
- a) Cuando no sabemos cuándo van a llegar los datos.
  - b) Cuando necesitamos tener resultados incrementales.
  - c) Cuando tenemos que procesar grandes cantidades de datos periódicamente y no necesitamos resultados de forma inmediata.
  - d) Cuando tenemos que procesar grandes cantidades de datos de forma asíncrona.
9. ¿Cómo definirías Hadoop?
- a) Es una plataforma para almacenar y procesar grandes cantidades de datos de forma distribuida, adecuada para procesos por lotes.
  - b) Es un sistema distribuido de almacenamiento de ficheros de gran tamaño.
  - c) Es un sistema distribuido de captura y análisis de grandes volúmenes de datos.
  - d) Es una plataforma para almacenar y procesar grandes cantidades de datos adecuada para tiempos de respuesta muy pequeños.
10. ¿Para qué tareas está optimizado Spark?
- a) Análisis exploratorio y entrenamiento y ajuste de modelos estadísticos con grandes cantidades de datos.
  - b) Captura y filtrado de grandes bloques de datos.
  - c) Consulta y agrupación de grandes cantidades de datos.
  - d) Visualización interactiva de grandes cantidades de datos.



## SOLUCIONARIO

---

1.	c	2.	a	3.	a	4.	a	5.	b
6.	d	7.	d	8.	c	9.	a	10.	a



## BIBLIOGRAFÍA

---

### Conceptos generales de Big Data

---

- "Big Data: Principles and best practices of scalable realtime data systems." Nathan Marz, James Warren. Manning Publications. 2015.
- "Big Data For Dummies". Judith Hurwitz, Alan Nugent et al. John Wiley & Sons. 2013.
- "Big Data For Beginners." Vince Reynolds. CreateSpace Independent Publishing Platform. 2016.

### Apache Hadoop

---

- Página principal del proyecto:  
<http://hadoop.apache.org/>

### Apache Spark

---

- Página principal del proyecto:  
<https://spark.apache.org/>

