INTELIGENCIA ARTIFICIAL

BIG DATA: APRENDIENDO CON MILLLONES DE DATOS

Structuralia

INDICE

1. INTRODUCCIÓN

En la última década se ha producido una revolución en la forma en la que los seres humanos interactuamos con la tecnología debido a la aparición de las redes sociales y los distintos tipos de Smartphone. Las redes sociales han cambiado la forma en la que los humanos interactuábamos entre nosotros mediante la utilización de la tecnología ofreciéndonos la posibilidad de crear diferentes tipos de contenidos y distribuirlos de forma sencilla, esto supuso la creación masiva de información mediante textos, comentarios e imágenes que alimentaba los contenidos de este tipo de aplicaciones y que era almacenado en los centros de datos de las grandes compañías propietarias de las aplicaciones. El uso de este tipo de aplicaciones fue potenciado con la aparición de los smartphones, los cuales permitían utilizar los nuevos tipos de teléfonos de forma similar a un ordenador incrementando a nivel exponencial el uso de este tipo de aplicaciones. Además, este nuevo tipo de dispositivos no sólo potenció la utilización de las redes, sino que produjo la aparición de centenares de nuevos tipos de aplicaciones que utilizaban las características que ofrecían como por ejemplo la posición GPS que permitía ofrecer funcionalidades basadas en la posición de los dispositivos. Las diferentes compañías comenzaron a almacenar ingentes cantidades de información con el fin de poder desarrollar técnicas que les permitieran analizarlas con el fin de aumentar las funcionalidades que podían ofrecer a sus usuarios y también incrementar mayores beneficios.

La utilización de toda esta información ha supuesto uno de los mayores desafíos tecnológicos para las organizaciones, tanto para los gigantes tecnológicos como para las pequeñas empresas, a la hora de encontrar un enfoque pragmático a la captura, análisis y utilización de la información sobre sus clientes, productos y servicios. Al comienzo la interacción con los clientes era sencilla ya que implica ofrecer un servicio simple y muchas veces a nivel local en sus propios mercados (países), pero según comenzó a evolucionar la tecnología de forma exponencial, los mercados se convirtieron en algo global despareciendo las barreras a nivel local lo que produjo que las empresas necesitaran crear algún tipo de ventaja competitiva con el fin de mantener a sus clientes y crecer con el fin de expandir sus servicios al nuevo mercado global en los que tenían que competir con aplicaciones y servicios similares que habían sido creados en diferentes mercados locales. Con el fin de poder ofrecer nuevas funcionalidades y servicios las empresas comenzaron a buscar manera de explotar de forma eficiente los datos existentes, mejorar los procesos de recolección de todos estos datos y construir sistemas que puedan utilizar toda está información con el fin de ofrecer ejecutar esas nuevas funcionalidades. En la **Figura 1** se presenta

una estimación de los datos que se generan por las aplicaciones más utilizadas en 2018 en sólo 60 segundos.

Figura 1: Cantidad de datos generados por minuto para las aplicaciones más utilizadas de Internet. Copyright © - @LoriLewis y @OfficiallyChadd

Como se puede observar en la **Figura 1**, la cantidad de información generada en tan sólo 60 segundos es extremadamente elevada. En 1 minuto se envían/reciben 187 millones de emails o se envían 38 millones de mensajes en WhatsApp o se realizan 3.7 millones de búsqueda en el buscador de Google o se envían 38 millones. Esto supone la generación de una gran cantidad de información que debe ser almacenada, procesada, analizada y utilizada con el fin de extraer

valor de los datos. Este proceso que puede considerar como el ciclo de vida de los datos y comienza con el llamado “Big Data” que es describe el gran volumen de datos, tanto estructurados como no estructurados, que son creados diariamente por la sociedad actual. Una vez generados estos datos deben almacenados, procesados, analizados y utilizados con el de completar el ciclo de vida de los datos y extraer algún tipo de valor a todos estos datos. En base a estas definiciones podríamos definir una serie de características básicas que definen el concepto de “Big Data”. Cuando el termino “Big Data” fue acuñado se definieron tres características o magnitudes básicas (Volumen, Velocidad y Variedad), conocidas como las 3Vs, que describían el significado de este termino, pero en mi opinión en necesario al menos incluir una cuarta magnitud referente a la Veracidad de la información que estamos utilizando.

- Volumen: El volumen se refiere a la cantidad de datos que son generados cada segundo. Es considerada como la característica más importante asociada al concepto de “Big Data”, ya que hace referencia a las cantidades masivas de información que se generan y almacenan con la finalidad de analizar todo esta información con el fin de obtener algún tipo de valor.
- Velocidad: La velocidad se refiere a la rapidez a la que es creada, almacenada y procesada la información en tiempo real. Actualmente existen muchos procesos en los que el tiempo es un elemento fundamental, como por ejemplo en los procesos de detección de fraude en las transacciones bancaria o la monitorización de ciertos eventos en redes sociales.
- Variedad: La variedad se refiere a al formato**, tipo y fuentes** de la información. Actualmente los datos tienen muchísimos tipos de formato desde los datos almacenados en las bases de datos hasta cualquier tipo de documentos de texto, correos electrónicos, datos de sensores, audios, vídeos, imágenes ,publicaciones en nuestros perfiles de redes sociales, artículos de blogs, las secuencias de clics que realizamos al navegar en una determinada página web, etc.; los cuales deben ser almacenados y procesados de forma similar con independencia del formato, el tipo y la fuente.
- Veracidad: La veracidad se refiere al grado de **incertidumbre de los datos**, es decir, al grado de fiabilidad que tiene la información que ha sido generado u obtenida. Actualmente se generan millones de datos por segundo, pero no todos tienen el mismo grado de veracidad. Los valores generados por un sensor perfectamente calibrado no son iguales que los tweet escritos por una persona que está intentando propagar información falsa

en las redes sociales, por lo que es muy importante definir sistemas que nos permitan identificar el grado de fiabilidad de la información.

2. EL CICLO DE VIDA DE LOS DATOS

Los datos son el elemento más importante alrededor del concepto de “Big Data” pero en nuestra opinión este concepto va más halla de la simple generación de datos de manera masiva. El “Big Data” puede describirse como proceso de recolección, almacenamiento y posterior análisis y manipulación de los datos a nivel masivo con el fin de extraer valor. Este proceso puede ser identificado como el ciclo de vida de los datos que puede ser descrito mediante el diagrama presentado en la **Figura 2**. Cada una de las fases de este proceso son descritas de forma detallada a lo largo de este documento.

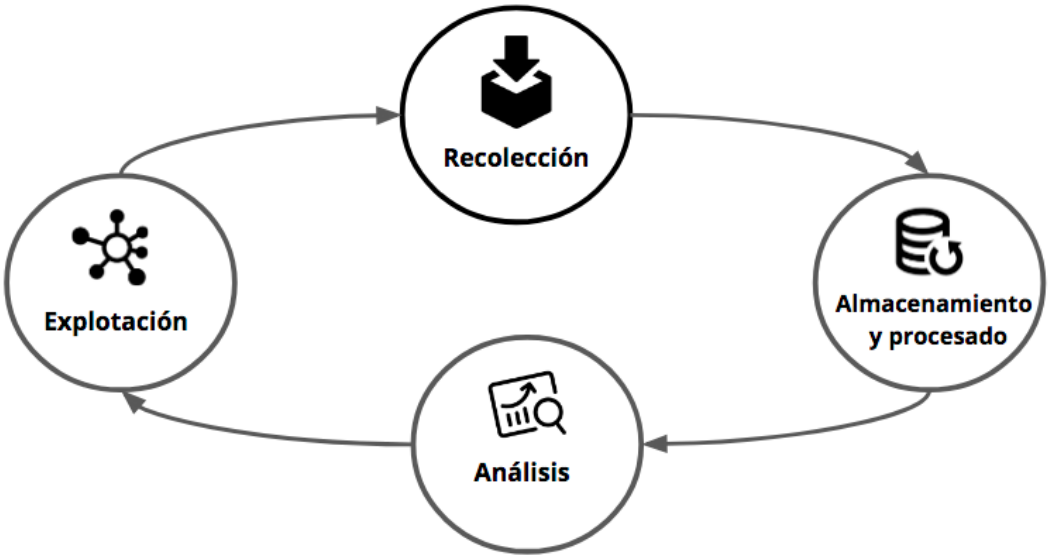

Figura 2: Ciclo de vida de los datos

3. RECOLECCIÓN DE LA INFORMACIÓN

El proceso de recolección de la información es la primera fase del ciclo de vida de los datos y probablemente la fase más importante de todas ya que dependiendo de la calidad y la cantidad de los datos que sean recolectados en esta fase se podrán aplicar las demás de forma correcta.

3.1 Tipos de datos

La información que se obtiene a través de diferentes tipos de sistemas y/o dispositivos puede clasificarse de diferentes manera en base a como estos son almacenados. Es decir, si los datos son almacenados en base a algún tipo de estructura o significado preciso que permite identificar su contenido de forma precisa. Es decir que la separación entre los tipos de datos se basa en parte en su significado o etiquetado.

3.1.1. Datos estructurados

El concepto de datos estructurados se refiere a aquel conjunto de datos que tienen una estructura, formato y longitud definida, siendo el formato más común de este tipo de datos una cadena de caracteres alfanuméricos, cómo por ejemplo, el nombre completo de un cliente o su dirección de facturación. Es decir, el significado preciso de cada dato ha sido correctamente definido por un humano y siempre guardan una estructura precisa. Es decir, la información del perfil de los usuarios de una página web, siempre está compuesta por un determinado número de campos, los cuales están siempre etiquetados con un determinado significado. Este tipo de datos han sido almacenados por las empresas desde la aparición de los primeros sistemas de almacenamiento y comúnmente se encuentra almacenados mediante la utilización de algún tipo de base de datos normalmente de tipo relacional. Se considera que este tipo de datos se corresponde con un 20% de la totalidad de datos que actualmente se tienen almacenados. Dependiendo del proceso que su utilice para su generación se pueden diferenciar dos tipos de datos estructurados.

Datos generados por computador

Los datos generados por computador son aquellos datos estructurados generados mediante una máquina sin ningún tipo de intervención humana. Algunos ejemplos de este tipo de datos estructurados son:

- Datos de funcionamiento o log: son los diferentes datos de funcionamiento o actividad generados por los diferentes servicios, aplicaciones, redes, etc. Este tipo de información es almacenada en fichero de log que suelen ser almacenados a nivel local en los dispositivos donde se ejecutan los servicios o aplicaciones. La información de log suele contener ingentes cantidades de información lo que suele suponer un coste muy elevado para las organizaciones debido a lo cual suele ser eliminado tras un periodo de tiempo.

A pesar de todo es una información muy útil que puede ser utilizada para identificar violaciones de seguridad o errores de ejecución.

- Datos de sensores: son datos generados por diferentes tipos de sensores (Acelerómetros, giroscopios, sistemas de identificación por radiofrecuencia, sistema de posición global, etc.) los cuales suelen incluir información referente al tipo de sensor y a la información obtenida por ellos. Por ejemplo, uno de los sensores más populares que se están utilizando actualmente son los dispositivos de identificación por radiofrecuencia (Radio Frequency Identification, RFID en sus siglas en inglés) [1] los cuales permiten almacenar información mediante la utilización de etiquetas o tarjetas que utilizan transpondedores RFID. En la **Figura 3** se presenta un ejemplo de una etiqueta RFID la cual está formada por un pequeño microchip que almacena la información del dispositivo y una antena transmisora que permite rastrear el dispositivo y extraer la información que ha sido obtenida por él. Además este tipo de etiquetas puede incluir otro tipo de sensores que permitan recolectar más información.

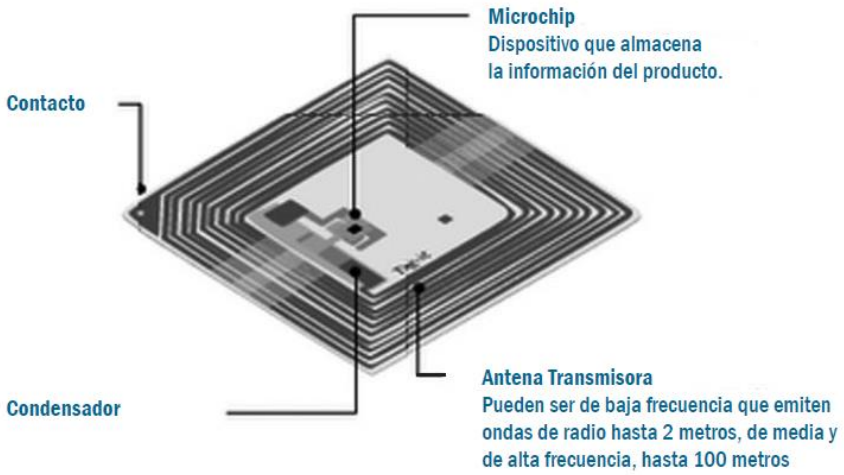

Figura 3: Ejemplo de un etiqueta RFID

Datos generados por humanos

Los datos generados por humanos son aquellos datos generados mediante la utilización de algún tipo de dispositivo y normalmente se corresponde con datos específicos por los usuarios, como por ejemplo sus datos personales, datos bancarios, etc. Algunos ejemplos de este tipo de datos estructurados son:

- Datos de entrada: este tipo de dato se corresponde con aquellos datos que son introducidos mediante algún tipo de interfaz. El ejemplo más típico de datos estructuras son aquellos recogido mediante los diferentes formularios de las aplicaciones móviles o web.
- Datos de transmisión de clics: este tipo de datos se generan cada vez que hace clic en un enlace en un sitio web. Durante muchos años han sido utilizados para identificar el comportamiento de los usuarios a la hora de interactuar con las aplicaciones y aplicar ciertos cambios con el fin de conseguir un mayor número de usuarios o cambiar el funcionamiento de las aplicaciones.
- Datos de ejecución en juegos: este tipo de datos se corresponde con acciones realizadas por los jugadores en los videojuegos online, con el fin de conocer la forma en la que jugamos los humanos y generar comportamientos más complejos para los agentes del juego.

3.1.2. Datos no estructurados

El concepto de datos no estructurados se refiere a aquel conjunto de datos que no siguen ningún formato específico, pero tienen algún tipo de realización debido a su formato de almacenamiento o proceso de recolección. Los datos no estructurados más comunes son las imágenes las cuales pueden representar cualquier tipo de contenido, pero son identificadas en base a su formato. Este tipo de datos pueden ser almacenados de diferentes formas, desde una base de datos hasta un conjunto de archivos en un directorio. Se considera que este tipo de datos se corresponde con un 80% de la totalidad de datos que actualmente se tienen almacenados. Muchas personas creen que el término datos no estructurado es un concepto erróneamente utilizado debido a que muchas veces los documento utilizados para generar este tipo de datos tienen una estructura. Por ejemplo, un documento de texto es considerado como un tipo de dato no estructura, pero posee una estructura bien definida está dividido en secciones que a su vez contienen párrafos que están formados por palabras que contienen letras. Pero realmente no sabemos cual es el significado de esos párrafos, o secciones o palabras. Es decir, no sabemos si se corresponden con una canción, un diálogo, un cuento, un comentario o cualquier otro cosa lo que los convierte en datos sin estructura sin significado preciso. Son los datos más comunes y los más difíciles de analizar ya que son datos de tipo general que normalmente no tienen una clasificación o etiquetado. Al igual que los datos de tipo estructurados estos pueden clasificarse en dos grupos dependiendo de si son recogidos mediante una máquina o un humano.

Datos generados por computador

Los datos generados por computador son aquellos datos de tipo no estructurado generados mediante una máquina sin ningún tipo de intervención humana. Algunos ejemplos de este tipo de datos no estructurados son:

- Imágenes de alta resolución: Esto tipo de datos se corresponde con imágenes recogidas por sistema de alta resolución como satélites meteorológicos o satélites de vigilancia militar. Este tipo de sistemas genera cientos de millones de imágenes, cada una de las cuales tiene un conjunto de información diferentes la cual es difícil de extraer y normalmente ha sido siempre analizada de forma manual por humanos.
- Datos científicos: Este tipo de datos se corresponde con el conjunto de datos referentes a magnitudes físicas que son recogidos por algún dispositivo de manera directa como pueden ser datos atmosféricos, física de alta energía, etc.
- Imágenes de baja resolución: Este tipo de imágenes se corresponde con información obtenida mediante sistema de seguridad tradicional, como cámaras de video vigilancia que permiten captar imágenes y videos.
- Datos de radar o sonar: Esto se corresponde con información global mediante vehículos que no son recogidos mediante ningún tipo de aplicación, datos de tipo meteorológico, datos de tipo sísmicos oceanográficos, etc.

Datos generados por humanos

Los datos generados por humanos son aquellos datos no estructurados generados por humanos, esta información no debe ser generadas mediante ningún tipo de interfaz específico. Algunos ejemplos de este tipo de datos estructurados son:

- Dato de texto: Este tipo de datos se corresponde con toda la información contenida en documentos físicos y/o electrónicos dentro de una organización. Por ejemplo, toda la documentación en papel anterior a las máquinas, encuestas, documentos legales, correos electrónicos, etc. Este tipo de información no sólo se encuentra almacenada en formato digital, sino también en formato físico, lo que está produciendo que muchas empresas comiencen a digitalizar todo esta información mediante sistemas de tipo automático o semi-automático.
- Contenido de aplicaciones: Estos datos se corresponden con toda la información generada mediante la diferentes aplicaciones que no tienen ningún tipo de estructura. La mayoría de ellos se corresponde con la información generada por las redes sociales y se

corresponde con cualquier dato sin estructura que incluye diferentes tipos de formato como por ejemplo imágenes, textos planos, emoticones, video, etc.

- Datos móviles: Este tipo de datos se corresponde con información producida por dispositivos móviles, siendo de este tipo las llamadas telefónicas en el caso de que estas sean grabadas por algún motivo, los mensajes de textos (SMS) y la información global de ubicación de los dispositivos móviles que se obtiene mediante la triangulación por antenas de telefonía.

3.1.3. Datos semi-estructurados

Los datos semi-estructurados son aquellos que no pueden catalogarse en ninguno de los grupos anteriores debido a que tienen algún tipo de estructura definida, pero es de tipo variable. Por ejemplo la estructura de una página web está formado por etiquetas basadas en el lenguaje HTML que tienen un significado preciso, pero su número, orden, contenido varia dependiendo de cada una de las páginas webs, por lo que a pesar de ser todos datos de tipo HTML, tienen una estructura parecida pero variable en base al contenido que presenten. Los formatos más comunes utilizados para definir tipos de datos semi-estructurados son los siguientes:

- El Lenguaje de Marcado para Hipertextos (HyperText Markup Language, HTML en sus siglas en inglés) [2] [3] es un lenguaje para la definición de la estructura básica de una página web. Se utiliza para definir el contenido de la página web. Este lenguaje es combinado con otros dos lenguajes para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). Este lenguaje está basado en una serie de etiquetas (<head>, <title>, <body>, <header>, <article>, <section>, <p>, , , <img>, etc) que permiten marcar o etiquetar los contenido de la página web de forma que serán mostrados de una manera específica en el navegador web.
- El lenguaje de marcado extensible (Extensible Markup Language, XML en sus siglas en inglés) [4] es un lenguaje de marcado de propósito general que utiliza etiquetas o tags dispuestos de forma jerárquica para identificar la estructura y significado de la información donde no existe un conjunto de etiquetas general. Es decir, las etiquetas utilizadas para representar la información contenida en un archivo XML es definida por los creadores de los ficheros. Este tipo de formato permite la creación de cualquier lenguaje basado en etiquetas. Es considerado como uno de los principales lenguajes utilizados para compartir información de tipo general entre diferentes aplicaciones web. Algunos ejemplos son XHTML, MathML, XSLT, RSS, y RDF. En la **Figura 4** se presenta un ejemplo de un fichero de tipo XML.

- El lenguaje de notación de objetos de JavaScript (JavaScript Object Notation, JSON en sus siglas en inglés) [5] es un lenguaje basado en la notación literal de objetos utilizada por el lenguaje de scripting JavaScript. Debido a su simplicidad y naturaleza se ha convertido en el actual estandar para la transferencia de información superando a XML en parte debido a que puede ser procesado (parseado) mediante la función eval de javascript que está presente en casi todos los navegadores web simplificando en el proceso de análisis sintáctico que tiene cualquier lenguaje. En la **Figura 4** se presenta un ejemplo de un fichero de tipo JSON.

XML

```
<empinfo>
   <employees>
      <employee>
         <name>James Kirk</name>
         <age>40></age>
      </employee>
      <employee>
         <name>Jean-Luc Picard</name>
         <age>45</age>
      </employee>
      <employee>
         <name>Wesley Crusher</name>
         <age>27</age>
      </employee>
   </employees>
</empinfo>
```

JSON

```
{  "empinfo" :
      {
            "employees" :  [
            {
                  "name" : "James Kirk",
                  "age" : 40,
            },
            {
                  "name" : "Jean-Luc Picard",
                  "age" : 45,
            },
            {
                  "name" : "Wesley Crusher",
                  "age" : 27,
            }
                              ]
      }
}
```

Figura 4: Ejemplo de la misma información representada mediante diferentes tipos de lenguajes.

3.2 Sistemas de recolección

En general el proceso de recolección de información es considerado como una fases más importantes y complejas del ciclo de vida de los datos. Además, muchas de las aplicaciones que se desarrollan actualmente tiene una acuciante necesidad de información para funcionar de manera correcta lo cual incrementa aún más la importancia del proceso de recolección. Existe múltiples formas de recolectar información para nuestras aplicaciones, pero en este tema sólo vamos a hablar de aquellas más comunes.

3.2.1. Métodos tradicionales

Los métodos tradicionales de recolección de información son aquellos que son utilizados de manera general y se corresponde con la utilización de archivos de logs, formularios o funcionalidades básicas para la recogida de la información. Este es la forma más común y suele funcionar perfectamente en aplicaciones que no tienen una necesidad de datos para funcionar correctamente, pero que una vez que los han recogido pueden ser utilizados para mejorar la experiencia de los usuarios u ofrecerles nuevas funcionalidades.

3.2.2. Web scraping

El web scraping [6] es una técnica de obtención de la información mediante la extracción del contenido de páginas web. Este tipo de técnica consiste en simular el proceso de navegación de un humano en una página web mediante un sistema automático, denominado robot, que descarga todo el contenido de cada una de las páginas web y a continuación utiliza los diferentes enlaces dentro del contenido para cargar nuevas páginas. Se puede considerar como un proceso recursivo dentro de cada dominio donde se comienza por la página principal (home) y se va navegando a través de los diferentes enlaces internos (links) de la página web de forma similar a como funcionaría un algoritmo de búsqueda donde el nodo raíz sería la página principal y los diferentes nodos sucesores serían los links de cada página. El proceso de web scraping está íntimamente relacionado por los sistema de indexación de contenidos utilizado por lo motores de búsqueda que realizan un proceso de crawling mediante la utilización de robots, denominados arañas, que recopilan información referentes a los enlaces presenten en las páginas webs.

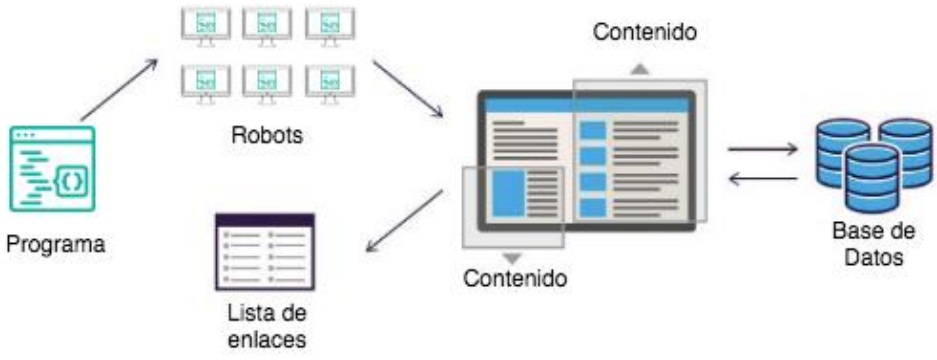

Figura 5: Funcionamiento básico de un web scraping.

Aunque normalmente el proceso de web scraping no sólo realiza un proceso de indexación, sino además que realiza una transformación sobre el contenido de la página web en datos

estructurados los cuales pueden ser almacenados y posteriormente analizados. En la Figura se presenta el funcionamiento básico de un web scraping, el cual consta de un programa o scraper (robot) que es desplegado en múltiples ordenadores cada uno de los cuales analizará un conjunto de páginas web definidas en la lista de enlaces. Este programa tendrá un sistema de procesamiento específico que extraerá cierta información de la página web que será combinada formando algún tipo de estructura y almacenada en algún tipo de base de datos. Este tipo de sistemas son muy eficaces y sencillos de construir pero puede incurrir en ilegalidades a la hora de recopilar información de páginas web con algún tipo de copyright o derechos de creación.

3.2.3. Colas de mensajes

El crecimiento exponencial del uso de ciertas aplicaciones hizo aparecer problemas en su rendimiento a la hora de procesar y almacenar la información introducida por los usuarios, ya que los servicios de inserción en las bases de datos no eran capaces de soportar el número de peticiones por segundos producidas por los usuarios. Con el fin de solventar este problema se desarrollaron las colas de mensajes, que son un sistema de comunicación de información asíncrona entre servicios (productor y consumidor) que se usa en arquitecturas de microservicios en las que no existe un servidor principal con el fin de evitar los posibles cuellos de botellas. Este tipo de sistemas de comunicación utilizan dos tipos de microservicios: un productor que inserta mensajes en la cola y un consumidor que extrae y/o eliminar mensaje de cola. De forma que los mensajes son almacenados en la cola de manera temporal hasta que son consumidos y eliminados de ella. La **Figura 6** muestra la estructura básica de una cola de mensajes.

Figura 6: Funcionamiento de una cola de mensajes

Este tipo de arquitectura permite crear sistema de recolección desacoplados mediante servicios de baja complejidad utilizando un búfer de almacenamiento temporal ligero de mensajes donde diferentes servicios escriben información (productores) que es procesada y transformada por otros servicios (consumidores) que insertan está información en las estructuras de almacenamiento. Este tipo de sistema de comunicación es utilizado de forma asidua para la

inserción de información masiva generada de forma manual o automática. Los dos sistemas de colas de mensajes más utilizados actualmente son las colas de mensajes Kafka [7] y las colas de mensajes RabbitMQ [8].

4. ALMACENAMIENTO DE LA INFORMACIÓN

El almacenamiento de la información es la segunda fase del ciclo de vida de los datos y consiste en almacenar la información de manera “física”. Es decir, en disco de forma que pueda ser manipulada de forma sencilla.

4.1 Ficheros de texto

Los ficheros de texto son el método tradicional para el almacenamiento de la información que simula el “funcionamiento de los libros” pero en formato digital. La estructura básica de un fichero de texto consiste en una secuencia de caracteres terminados por una marca única de fin de fichero. Los caracteres a su vez se encuentran divididos en fragmentos separados mediante marcas que señalan los finales de línea del fichero. Cada una de estas líneas puede considerarse como una pieza básica de información. Este es el formato mediante el cual está almacenada toda la información en los ordenadores y en el caso de Big Data se suele utilizar para el almacenamiento de los ficheros de información y logs de las aplicaciones, servidores, bases de datos.

4.2 Bases de datos

Una base de datos [9][10] es un “almacén” que permite almacenar grandes cantidades de información de forma organizada que luego puede ser consulta de forma sencilla. De manera más formal, se puede definir una base de datos como un sistema compuesto por un conjunto de datos almacenados en disco que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto de datos. Dependiendo de cómo estos datos se encuentran relacionados entre sí, pueden distinguirse dos tipos de bases de datos.

4.2.1. 4.2.1 Bases de datos relaciones

Una base de datos relacional [11] (Structured Query Languaje, SQL en sus siglas en inglés) es un conjunto de datos estructurados entre los cuales existen una serie de relaciones predefinidas. Los datos se organizan mediante un conjunto de tablas que representan las diferentes entidades que serán definidas en la base de datos, formadas por un conjunto de filas y columnas. Donde cada columna representa un tipo de datos o atributos de la tabla y cada fila se corresponde con

un ejemplo con valores específicos para cada una de los atributos. Cada una de estas filas se encuentra identificada de manera univoca por un atributo especial, denominado clave principal. Además las tablas se relacionan entre sí mediante la utilización de atributos, denominados claves ajenas, que toman el valor de la clave principal de otra tabla.

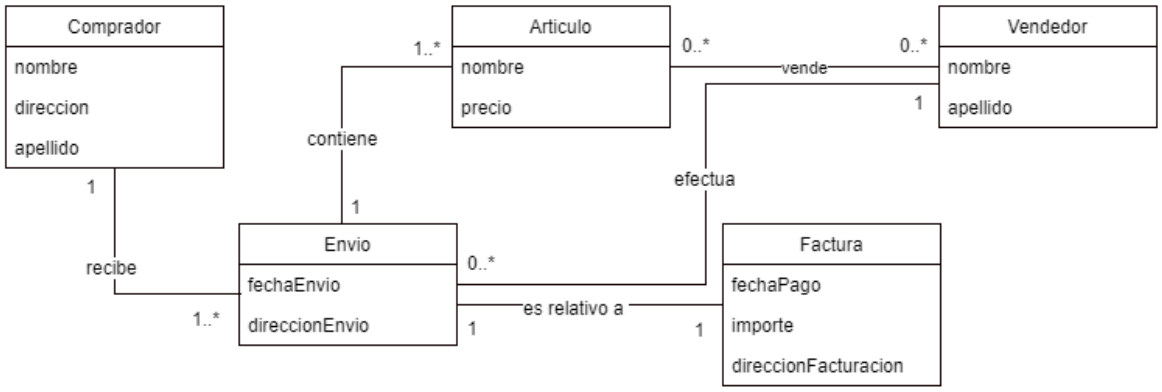

Figura 7: Ejemplo de una base de datos relacional para la compraventa de artículos

Para representar la estructura de cada una de las tablas e interactuar con ellas mediante la utilización operaciones (consultas, procedimientos almacenados, triggers, etc.) se utiliza el lenguaje de consulta estructurada [12] (Structured Query Languaje, SQL en sus siglas en inglés) que es soportado por todos los motores de bases de datos relaciones. Este lenguaje permite definir la estructura de la tabla y sus relaciones, así como las diferentes operaciones de inserción, eliminación, actualización y búsqueda. Algunos ejemplos de bases de datos relaciones son:

- MySQL es un sistema de gestión de bases de datos relacionales de código abierto (Relational Data Base Management System, RDBMS en sus siglas en inglés) y es considerada como la base de datos más popular del mundo para entornos de desarrollo web.
- PostgreSQL es un sistema de gestión de base de datos relacional orientado a objetos de código abierto (Object Data Base Management System, ODBMS en sus siglas en inglés). Además de las funcionalidades básicas del sistema de gestión de base de datos permite ejecutar procedimientos almacenados en diferentes lenguajes de programación.

- Oracle es un sistema de gestión de base de datos relacional de tipo objeto-relacional (Object-Relational Data Base Management System, ORDBMS en sus siglas en inglés) para su uso en servidores empresariales.
- Microsoft SQL server es un sistema de gestión de base de datos de tipo relacional desarrollado por la empresa Microsoft. Existen diferentes versiones denominadas Express, Web, Standard y Enterprise.
- Amazon Aurora es un sistema de gestión de base de datos relacional compatible con MySQL y PostgreSQL desarrollado por la compañía Amazon.
- MariaDB es un motor de base de datos compatible con MySQL y derivado de MySQL, que está siendo desarrollado por los desarrolladores originales de MySQL.

4.2.2. Bases de datos no relacionales

Una base de datos no relacional [13] (Not only Structured Query Languaje, NoSQL en sus siglas en inglés) es un conjunto de datos estructurados con esquemas flexibles donde no existe ningún tipo de relación implícita entre los diferentes esquemas que representan los datos. Este tipo de bases de datos comenzaron a utilizarse a finales de la década del 2000 debido a los problemas de rendimiento que presentaban las bases de datos relacionales cuando el número de datos almacenados era muy elevado. Este tipo de bases de datos no siguen un único modelo de representación como ocurría en las bases de datos relacionales, sino que existen diferentes tipos:

- Documentales: Son bases de datos que almacenan datos semi-estructurados en forma de documentos cada uno de los cuales puede tener una estructura diferente. Lo que permite almacenar la información de manera más natural. Algunos ejemplos de bases de datos de este tipo son: IBM Lotus Domino, MongoDB o SimpleDB.
- En grafo: Son bases de datos que almacenan datos estructurados mediante un grafo, donde los nodos representan las entidades de información y las aristas las relaciones que existen entre los nodos. Este tipo de estructura permiten utilizar la teoría de grafos pare recorrer la base de datos. Algunos ejemplos de bases de datos de este tipo son: Neo4j, AllegroGraph, ArangoDB o InfiniteGraph.
- Clave-valor: Son bases de datos que almacenan datos semi-estructurados mediante un método de clave-valor, donde la clave se corresponde con un identificador unívoco y el valor con un conjunto

de información. Tanto las claves como los valores pueden ser cualquier cosa, desde una simple cadena de caracteres hasta un objeto complejo. Algunos ejemplos de bases de datos de este tipo son: DynamoDB, Apache Cassandra o Redis.

En este tipo de base de datos (NoSQL) la información generalmente se almacena como un documento JSON. Es decir, la información se almacena mediante atributos/propiedades en un solo documento identificado mediante una clave unívoca, lo que permite a este tipo de bases de datos gestionar grandes cantidades de información independiente.

4.3 HDFS

El Hadoop Distributed File System (HDFS) es un sistema de ficheros distribuido, escalable y portable para el almacenamiento y manipulación de ficheros de gran tamaño donde el tamaño de bloque de los ficheros (64 Mb) es muy superior a los sistemas de ficheros empleados por los sistemas operativos (512 bytes, 1 Kb, 2 Mb) con el fin de minimizar el tiempo en los proceso de lectura. La información almacenada en este sistema de ficheros sigue el patrón “Write once read many” (escribir una vez y leer muchas veces) donde los ficheros son normalmente escritos una única vez mediante algún tipo de proceso batch y son leídos múltiples veces mediante diferentes algoritmos de análisis. Este tipo de distribución de información se consigue mediante una arquitectura distribuida donde los ficheros son divididos en bloques que son distribuidos en entre los diferentes nodos de la arquitectura (los bloques de un mismo fichero pueden ser almacenados en diferentes nodos). Un sistema de ficheros de tipo HDFS está compuesto por un clúster de nodos que almacenan la información de manera distribuida mediante una arquitectura de tipo Maestro-Esclavo formada por dos tipos de nodo:

- Nodo de Datos (DataNode): Son cada uno de los nodos en los que se almacenan los bloques de información de los diferentes ficheros los cuales son recuperados bajo demanda del nodo maestro. Estos son los nodos de tipo esclavo de la arquitectura debido a que ejecutan las ordenes de almacenamiento y recuperación definidas por el nodo de nombres. Con el fin de ofrecer alta disponibilidad los bloques de datos son replicados en múltiples nodos de datos (por defecto en 3 nodos).
- Nodo de Nombres (NameNode): Es el nodo raíz del clúster que se encarga de gestionar todo el referente a los nodos de datos. Posee un espacio de nombres para identificar a cada uno de los nodos de datos de forma unívoca y se encarga de gestionar la distribución de la

información identificando la localización de los diferentes bloques de los ficheros almacenados. Sólo existe un nodo de este tipo en el clúster HDFS.

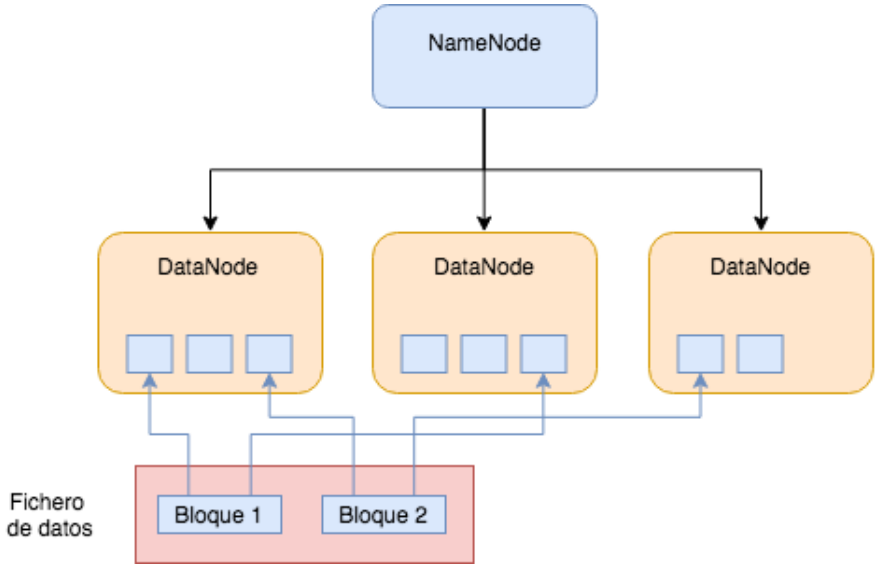

Figura 8: Estructura básica de una clúster Hadoop de 4 nodos

En la **Figura 8** se presenta la arquitectura básica de un clúster hadoop formada por 4 nodos: 1 NameNode y 3 DataNodes. Además se describe de forma gráfica como se realiza la distribución de los bloques de un fichero de datos. En este caso cada bloque se almacena en dos nodos.

5. ANÁLISIS DE LA INFORMACIÓN

Una vez que la información ha sido recolectada, procesada y almacenada en los diferentes sistemas de almacenamiento, es posible explotar esta información con el fin de extraer algún tipo de conocimiento sólo sea posible obtener y/o calcular mediante la combinación de la información a nivel global o parcial. Este tipo de procesos de explotación de grandes cantidades de información comenzaron a desarrollarse en la de década de los 2000 por empresas como Google, la cual tenía la necesidad de procesar grandes cantidades de información para calcular el PageRank de las páginas web cuya cantidad había crecido exponencialmente. El proceso de calculo del PageRank necesita realizar operaciones de multiplicación sobre matrices de gran tamaño con el fin de calcular el valor para cada página web, lo cual era prácticamente imposible realizar en un entorno tradicional debido a la gran cantidad de información que debía ser utilizada. Con el fin de resolver este tipo de problemas surgió el modelo de programación MapReduce [14] que permitía realizar operaciones matemáticas sobre volúmenes de información muy grandes

mediante la utilización de la paralelización de operaciones independientes. Este modelo de computación demostró ser extremadamente útil para la resolución de problemas, que debido a su complejidad computacional o al tamaño de la información que utilizaban, no podían ser resueltos mediante las técnicas tradicionales. La facilidad para resolver nuevos problemas mediante este nuevo “paradigma” produjo una rápida diseminación de este tipo de modelo de programación dando lugar a la aparición de diferentes frameworks de programación que permitían desplegar, de forma muy sencilla, algoritmos en pequeños clústeres de ordenadores internos. Esto produjo la aparición de nuevas tecnologías relacionadas con la creación, recolección y manipulación de información así como la utilización de algoritmos de Aprendizaje Automático sobre conjuntos de información muy grandes permitiéndoles demostrar su gran potencial para crear modelos de aprendizaje.

5.1 Programación paralela mediante MapReduce

MapReduce [14] es un modelo de programación fuertemente orientado a la ejecución paralela y distribuida entre múltiples ordenadores, que permite trabajar con grandes volúmenes de información de manera que dado un conjunto muy grande de información de entrada (de un archivo, de una base de datos o cualquier otra fuente de datos) es capaz de generar un conjunto de datos de salida, de tamaño indeterminado, mediante la combinación o manipulación de la información de entrada utilizando dos funciones de programación funcional muy conocidas: las funciones Map y Reduce.

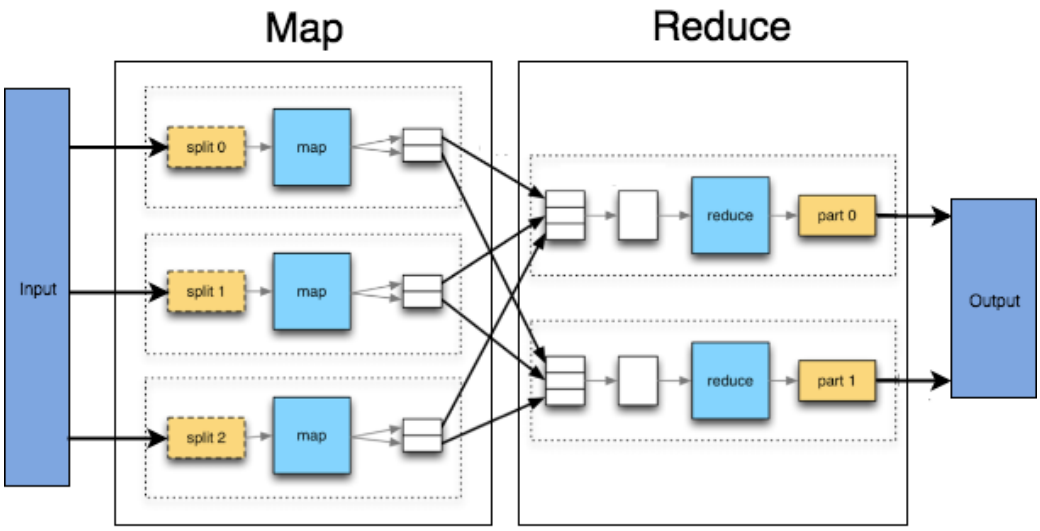

Figura 9: Funcionamiento de MapReduce

En la **Figura 9** se presenta una descripción gráfica sobre como funciona MapReduce. Este proceso se compone de dos fases (en realidad son cuatro, pero la primera y la tercera fases son transparentes para el desarrollador). La fase de división (Splitting), la fase de Mapeo (Map), la fase de agrupamiento (Shuffling) y la fase de reducción (Reduce). La fase de división divide el conjunto de entrada en fragmentos que pueden ser procesados de manera individual en ordenadores diferentes. La fase de agrupamiento consiste en agrupar la información generada por el proceso de mapeo en base a una clave con el fin de que puedan ser utilizados por el proceso de reducción. Este proceso fue diseñado para la utilización de datos estructurados en forma de tuplas del tipo (clave, valor), de manera que tanto la información de entrada como la información de salida es representada mediante este formato.

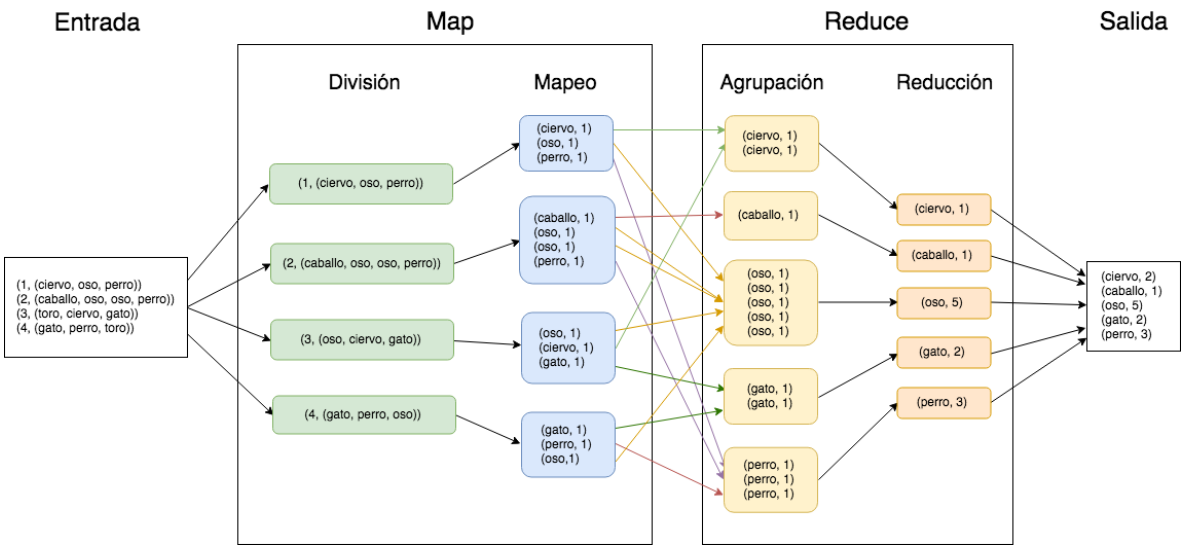

Figura 10: Ejemplo de funcionamiento de MapReduce para contar el número de ocurrencias de cada palabra.

5.1.1. La Función Map

La función Map es una función de mapeo que recibe como entrada una tupla en la forma (clave, valor) para generar como salida una lista de tuplas en la forma (clave, valor) donde las claves y valores de entrada son diferentes a los de entrada. Dado un conjunto de tuplas de entrada, la función map realiza una división de los elementos del conjunto aplicando sobre cada uno de ellos una operación que da lugar a una lista de tuplas de salida. La gran ventaja del proceso de mapeo es que puede ser paralelizado en diferentes procesos con el fin de minimizar el tiempo de mapeo. En el ejemplo presentado en la **Figura 10** se aplica un proceso de MapReduce para contar el número de ocurrencias de cada palabra en un conjunto de frases. Para ellos si tiene como

entrada un conjunto de frases (cada una se corresponde con una tupla en entrada), sobre las cuales se aplica la función map de manera individual sobre cada una de las frases del conjunto de entrada. En general el proceso de mapeo consiste simplificar la semántica de la información

5.1.2. La Función Reduce

Una vez ejecutada la operación de map se aplica la operación Reduce, que realiza una agrupación de todos las tuplas procedentes de las diferentes listas generadas por la operación Map que contienen la misma clave creando un conjunto de listas donde todos los valores comparten la misma clave. A continuación estas listas son reducidas mediante algún tipo de operación que disminuye el tamaño de los conjuntos para finalmente combinarlas todas ellas para general la salida del proceso de MapReduce. En el ejemplo presentado en la Figura se realiza una agrupación de las palabras similares de cada una de las listas generados por la operación Map y a continuación se realiza una operación que calcula la suma de los valores de las tuplas de cada conjunto con el fin de conocer el número de ocurrencias de cada palabra. Para finalizar todas las listas son combinadas en una lista final que se corresponde con el resultado de la operación.

5.1.3. Frameworks basados en MapReduce

Debido a la potencia del modelo de programación paralela de MapReduce comenzaron a desarrollarse diferentes frameworks de desarrollo que permitieran implementar de manera sencilla algoritmos que utilizasen este modelo de funcionamiento. Los primeros algoritmos implementados sobre estos frameworks consistían en operaciones matemáticas básicas, pero según se fue estandarizando el uso de este tipo de framerworks se comenzaron a incluir muchos de los algoritmos de Aprendizaje Automático descritos en los dos capítulos anteriores, con el fin de producir modelos de aprendizaje sobre grandes conjuntos de datos. Los frameworks más utilizados basados en MapReduce son:

- Apache Hadoop: Es un framework [15] para la creación de aplicaciones distribuidas basado en MapReduce que almacena la información en memoria física (Disco) mediante la utilización del sistema de archivos HDFS.
- Apache Spark: Es un framework[16] para la creación de aplicaciones distribuidas basado en MapReduce que almacena la información en memoria volatil (RAM). Normalmente este framework se apoya en algún tipo de almancenamiento físico del que obtiene la información de entrada y donde almacena la información de salida.

El despliegue de este tipo de tecnología suponía la utilización de complejas infraestructuras de grandes clústeres de ordenadores que permitieran distribuir el procesamiento de las operaciones y el almacenamiento de los datos.

5.2 5.2 La computación paralela en la nube

Al inicio toda la infraestructura necesaria para el despliegue de este tipo de tecnologías, basadas en MapReduce, se realizada mediante redes de ordenadores “locales”. Es decir las diferentes organizaciones tenían sus propias redes de ordenadores privadas en las cuales ejecutaban sus procesos paralelos. Esto a larga suponía un coste económico muy elevado para las organizaciones, ya que tenían que realizar una inversión en infraestructura y personal para disponer de una red de ordenadores que en muchos casos no se utilizaba de forma completa debido a que no se estaban ejecutando procesos paralelos las 24 horas de cada día. Este importante inconveniente hacía que la mayor parte de las compañías no pudiera acceder a este tipo de tecnología debido a sus elevados costes. La necesidad de infraestructuras no sólo de computo, sino de almacenamiento o de despliegue de servidores o servicios supuso una gran oportunidad de negocio para los grandes gigantes tecnológicos los cuales disponían de grandes infraestructuras que podían ofrecer a las pequeñas y medianas empresas servicios para el almacenamiento y manipulación de datos en “la nube” lo que permitía a las empresas alquilar la infraestructura necesaria pare desplegar sus procesos de forma temporal a bajo coste.

La computación en “la nube” (Cloud computing) es un paradigma que permite proporcionar un conjunto de servicios compartidos (aplicaciones, almacenamiento, servidores, plataformas de despliegue, etc.) a través de una red común, que normalmente es Internet. Este tipo de paradigma permitió una democratización del “Big Data” ya que cualquier empresa podía acceder a este tipo de servicios de manera muy barata sin necesidad de desplegar una infraestructura física propia. A pesar de las grandes ventajas que ofrecía este tipo de paradigma muchas empresas han decido no adoptarlo de forma completa, en parte debido al tipo de información que manipulan. Por ejemplo, los servicios de salud publica almacenan información personal de todos sus pacientes cuya privacidad debe ser mantenida en base a una serie de leyes estatales, lo cual hace que este tipo de información deba ser almacenada en base a una serie de criterios con el fin de mantener su privacidad. Este tipo de situaciones hizo que surgieran variaciones en este paradigma dando lugar a diferentes modelos de implementación de la nube:

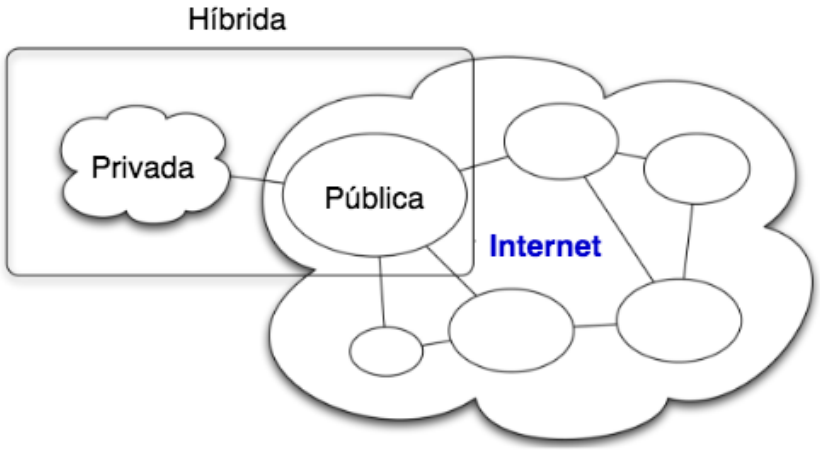

Figura 11: Modelos de implementación de la nube

- Las nubes privadas son una infraestructura bajo demanda gestionada para un solo cliente o compañía que tiene control total sobre la nube. Es decir, la empresa es propietaria de toda la infraestructura física y tiene el control total sobre el acceso y utilización de sus recursos (servidores, red y disco). La gestión de este tipo de nube puede ser gestionada mediante una empresa externa a la compañía propietaria de la nube, pero esta tiene control total sobre el acceso a los servicios. Este tipo de nube permite mantener la privacidad de la información almacenada en la nube, pero suele tener un coste más elevada ya que la infraestructura tiene que ser mantenida por la propia empresa.
- Las nubes públicas son una infraestructura mantenida y gestionada por un proveedor de servicios el cual no tienen ningún tipo de vinculación con la compañía y/o usuarios que acceden a los servicios que ofrece la nube. Normalmente, existe algún tipo de vinculación contractual ya que el proveedor alquila sus servicios a lo usuarios. En este tipo de nube la información y las aplicaciones de los diferentes usuarios (clientes) se encuentran distribuidos de manera compartida entre los componentes físicos de la arquitectura sin tener ningún conocimiento acerca de los otros usuarios que están utilizados los servicios. Este es el caso más común en la nube debido a su escalabilidad, donde una compañía pone al servicio de los clientes una infraestructura bajo demanda del usuario a un determinado coste, el cual es bastante bajo ya que el mantenimiento y gestión de la infraestructura es transparente para el usuario final.
- Las nubes híbridas son una combinación de las dos anteriores donde los diferentes usuarios son propietarios de una parte de la nube y comparten otras de una manera controlada. Este

tipo de nubes intentan ofrecer lo mejor de ambos casos pero tiene un gran problema relacionado con la seguridad y el acceso a varios tipos de nubes a la vez, lo que aumenta su complejidad a la hora de implementaras. A pesar de sus problemas son bastante útiles para el despliegue de aplicaciones sencillas, que no requieran de ninguna sincronización o necesiten almacenamiento de información con diferentes niveles de privacidad.

Independientemente del modelo implementación de la nube esta puede ofrecer un conjunto de servicios, como se presenta en la **Figura 12**, que se distribuyen en diferentes niveles que van desde la ejecución de aplicación alojadas en la nube hasta el control total de la infraestructura física de los servidores que componen la nube. Dependiendo de que nivel de control se tenga sobre los diferentes elementos de la nube existen tres tipos diferentes de modelos.

Figura 12: Diferentes tipos de modelos de servicios ofrecidos por la nube

- El software como servicio (Software as a Service, SaaS en sus siglas en inglés) es el más limitado de los modelos de servicio ofrecidos por la nube. Este modelo permite a los usuarios acceder y utilizar un conjunto de aplicaciones alojadas en la nube a través de Internet. Este tipo de modelo normalmente se basa en la utilización de un navegador web para acceder al servicio donde toda la infraestructura subyacente, el middleware, el sistema operativo, el software y los datos de las aplicaciones se encuentran almacenados y administrados por el proveedor del servicio, el cual garantiza la disponibilidad y la seguridad de la aplicación y de sus datos. Este tipo de modelo suele ser el más sencillo y económico ya que provee de una serie de servicios básicos sin necesidad de preocuparse del mantenimiento, despliegue o desarrollo de los elementos que hacen que los diferentes servicios funcionen correctamente. Un ejemplo de este tipo de modelo son los sistemas de mensajería alojados en la nube proporcionados por Microsoft o Google.

- Plataforma como servicio (Platform as a Service, PaaS en sus siglas en inglés) es un modelo más complejo del anterior que provee de un entorno de desarrollo e implementación completo en la nube que permite desplegar desde aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales mucho más complejas desarrolladas por la propia compañía que contrata el servicio. El modelo PaaS permite configurar el middleware, las herramientas de desarrollo y test, los diferentes servicios de inteligencia empresarial (BI), los sistemas de administración de bases de datos, etc. Es decir, este modelo permite controlar el ciclo de vida completo del software de una aplicación: compilación, pruebas, implementación, administración y actualización. Este tipo de modelo permite a la empresa que contrata el servicio despreocuparse de todo del proceso de adquisición y pago de licencias software de las diferentes herramientas de desarrollo o despliegue así como de la infraestructura física en la cual se va a desarrollar ese servicio. Es decir, este tipo de modelo permite adquirir servidores que se encuentran desplegados en la nube, donde el usuario tiene el control parcial del servidor, pero no debe gestionar nada referente al sistema operativo o la arquitectura física. Un ejemplo de este tipo de modelo son los servidores que se pueden adquirir a través de Google Cloud.
- La infraestructura como servicio (Infrastructure as a Service, IaaS) es el modelo más complejo y consiste en proveer una infraestructura informática inmediata que se aprovisiona y administra a través de Internet. Es decir, es una forma de adquirir un servidor sobre el que se tiene control total pero sin asumir la compra y administración de los servidores físicos ya que el proveedor del servicio administra la infraestructura física y la seguridad, mientras que el usuario administra la infraestructura software ya que se encarga de instalar, configurar y administrar su propio software (sistemas operativos, middleware y aplicaciones).

5.3 Analizando los datos

Todos estos de modelos de implementación de nubes, modelos de servicios y modelos de programación son las herramientas que nos permiten construir el software que puede analizar la gran cantidad de datos que tenemos almacenados en los diferentes sistemas de almacenamiento. Todas estas herramientas nos permitirán analizar la información en base a dos estrategias:

- Generación de modelos: La generación de modelos se basa en utilizar técnicas de Aprendizaje Automático, cómo las descritas en los temas 5 y 6 de este curso, sobre

grandes volúmenes de información con el fin de construir modelos de aprendizaje que puedan permitir a nuestras aplicaciones desplegar sistemas de toma de decisiones automáticos con el fin de mejorar la experiencia de los usuarios y/o predecir información que pueda ser utilizado por los usuarios para interactuar de forma diferentes con los servicios ofrecidos.

- Inteligencia de negocio: La inteligencia de negocio (Business Intelligence, BI en sus siglas en inglés) se basa en la generación de conocimiento mediante el análisis de grandes volúmenes de información con el fin de presentar esta información de manera más compacta facilitando el proceso de toma de decisiones de los seres humanos. Es decir, este tipo de estrategias busca presentar toda la información disponible con el fin de presentarla de manera visual mediante gráficas, imágenes, diagramas con el fin de reflejar aquellos elementos más importantes de forma que puedan facilitar el proceso de toma de decisiones de los humanos. Por ejemplo, con el fin de aumentar la inversión en un determinado tipo de productos, contratar más personal para reforzar alguna de las áreas de la compañía, etc.

6. EXPLOTACIÓN DE LA INFORMACIÓN

Esta es la última fase del ciclo de vida de los datos y consiste en explotar los diferentes productos que han sido generados tras todo el proceso anterior. Es decir, buscar una manera de utilizar los diferentes modelos de aprendizaje o el conocimiento que han sido generado en la fase de análisis con el fin de mejorar algunos aspectos de los productos que se ofrecen o del funcionamiento de la compañía. Normalmente el proceso de explotación de estos productos se suele realizar al menos de tres maneras diferentes.

- Interfacez de programación de aplicaciones: Los diferentes modelos de aprendizaje que son producidos durante la fase de análisis pueden ser integrados en nuestros productos de manera sencilla mediante la utilización de un API. Un interfaz de programación de aplicación (Application Programming Interface, API en sus siglas en inglés) es un conjunto de método o funciones que ofrece una biblioteca o servicio que puede ser utilizado por otro servicio. Los modelos pueden definirse como funciones, que hemos aprendido, donde dada una entrada definida en un determinado formato son capaces de generar una salida definida en otro formato. Por ejemplo, si hemos conseguido construir un modelo que nos indique el precio de una vivienda en España seremos capaces de obtener un

precio si suministramos a nuestro modelo la información de una vivienda. En base a todo esto nuestros modelos de aprendizaje pueden ofrecidos como un servicio localizado en nuestra nube pública o privada que ofrece una API a la que pueden acceder otros servicios.

- Visualización de datos: Existen muchas situaciones en la cuales debido a la gran cantidad de información o su complejidad es muy difícil extraer conclusiones sencillas de los datos. Una forma muy útil de conseguir extraer estas conclusiones es mediante una representación visual de los datos que puede permitirnos resaltar ciertas características que no son visibles sin la suficiente cantidad de datos combinados o si estos no son presentados de manera visual. Un ejemplo de un sistema de visualización de datos es la aplicación Circos to Genomics que permite comparar genomas secuencialmente y plasmar la relación entre ellos en pares visualmente. Circos es un software para visualización de datos que permite representar mediante un modelo circular en forma de anillo los 24 cromosomas del genoma humano (incluidos los cromosomas sexuales X e Y) y posicionarlos con genes relacionados con ciertas enfermedades colocados fuera del círculo. En la **Figura 13** se muestra un ejemplo de este tipo de visualización donde para cada cromosoma presenta su mapa de genes donde los datos colocados en la parte superior del anillo cromosómico resaltan los genes implicados en ciertas enfermedades como el cáncer, la diabetes y el glaucoma. Los datos colocados dentro del anillo vinculan los genes relacionados con una determinada enfermedad que se encuentran en la misma ruta bioquímica (gris) y el grado de similitud para un subconjunto del genoma (coloreado).

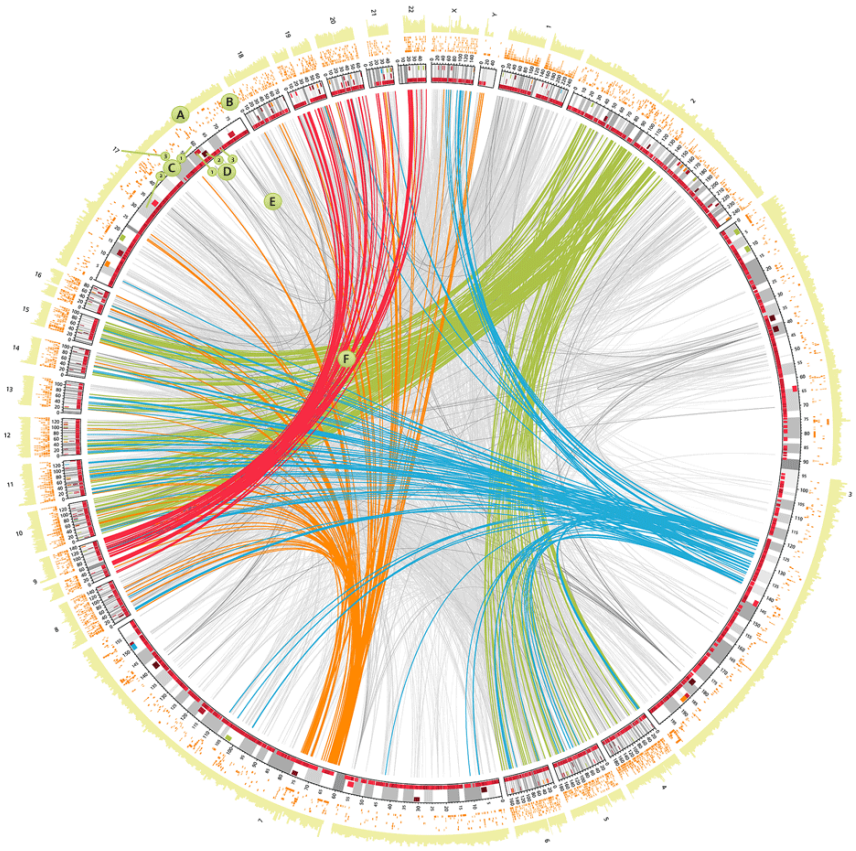

Figura 13: Representación de las relaciones entre los diferentes genes con diferentes tipos de enfermadas.

- Cuadros de mando: Un cuadro de mando es la representación simplificada de un conjunto de indicadores que dan una idea general del comportando de un área o un proceso. Es decir, permiten representar gráficamente la tendencia o el estado de un conjunto de indicadores (Key Performance Indicator, KPI en sus siglas en inglés) considerados relevantes para la gestión del proceso o área que se está analizando. La idea es visualizar de forma sencilla todos los indicadores (KPI) comparándolos con sus respectivos valores objetivos (Key Goal Indicator, KGI en sus siglas en inglés). Esta forma de visualización permite obtener una visión global del estado del proceso o área con el fin de identificar aquellas acciones necesarias para cambiar y/o mantener los resultados presentados en el cuadro mando. Además es muy importante que los cuadros de mando ofrezcan funcionalidades extra que permitan obtener análisis con detallados y realizar consultas más específicas sobre los datos. La **Figura 14** presenta un ejemplo de un cuadro de mando. Algunos de los sistemas de BI más utilizados actualmente son Tableau [17], Looker [18] o Microsoft Power BI [19].

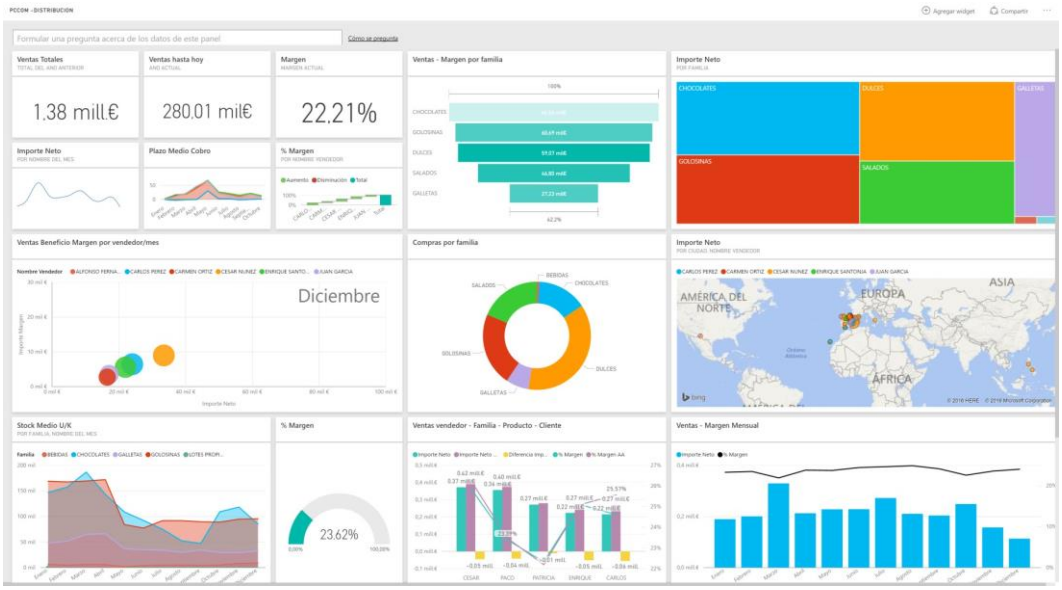

Figura 14: Ejemplo de un cuadro de mando

7. CONCLUSIONES

En este tema se ha presentado de forma sencilla el concepto de “Big Data” y en que consiste el ciclo de vida de los datos. A lo largo de este tema se ha intentado presentar la importancia de la información que es nuestra opinión el concepto más importante en “Big Data” debido a que la gran cantidad de información que los humanos generamos en cada segundo es lo que ha creado este concepto, termino o ciencia. No está del todo claro como va a influenciar nuestras vidas el “Big Data” en los próximos años y es algo a tener en cuenta. La gran ventaja del “Big Data” es que es capaz de extraer información referente a las decisiones mayoritarias que tomamos los humanos, es decir, es capaz de identificar los comportamientos de la mayoría de la población que genera información. Esta información puede ser muy útil con el fin de eliminar comportamientos nocivos de nuestra sociedad pero también puede ayudar a ciertas compañías a manipular nuestros comportamientos introduciendo ciertos estímulos que les permitan modificar esos comportamientos hacia un resultado esperado con el fin de obtener un mayor beneficio. Por lo que el gran reto al que se enfrenta el “Big Data” en estos momentos es como identificar cuales son los límites a la hora de utilizar toda la información que los humanos generamos continuamente y si los sistemas construidos mediante el uso de esta información no más que mejor sistemas que a larga van a sugestionarnos para elegir sólo entre aquellas

opciones que son validas desde el punto de vista de el ente que controla los datos. A pesar de todo, el “Big Data” se convertirá en uno de los recursos más útiles de la humanidad ya que nos permite ver de forma “clara” entre un inmenso universo de datos.

8. REFERENCIAS

[1] 1 Dargan G., Johnson B., Panchalingam M. and Stratis C. (2004), The Use of Radio Frequency Identification as a Replacement for Traditional Barcoding, IBM.

[2] Duckett J. (2011), HTML & CSS: Design and Build Web Sites, John Wiley & Sons Inc. ISBN 9781118008188

[3] Lenguaje HTML (World Wide Web Consortium): https://www.w3.org/html/

[4] Lenguaje HTML (World Wide Web Consortium):https://www.w3.org/XML

[5] https://www.json.org/

[6] Vanden Broucke S. and Baesens B. (2018), Practical Web Scraping for Data Science: Best Practices and Examples, Apress. ISBN 978-1-4842-3582-9

[7] Kafka: https://kafka.apache.org

[8] RabbitMQ: https://www.rabbitmq.com

[9] A. de Miguel y M. Piattini. (1990) Fundamentos y Modelos de Bases de Datos. (2ª edición). RA-MA. ISBN 9788478973613

[10] Silberschatz A., Korth H. and Sudarshan S. (2014). Fundamentos de Bases de Datos. McGraw-Hill Interamericana de España S.L. (6ª edición). ISBN 978-8448190330

[11] M. Piattini, E. Marcos, C. Calero y B. Vela. (2006) Tecnología y Diseño de Bases de Datos. RA-MA. ISBN 9788478977338

[12] Godoc E. (2014). SQL – Fundamentos del lenguaje, ENI. ISBN 9782746091245.

[13] Sullivan D. (2015). NoSQL for Mere Mortals, Addison-Wesley. ISBN 9780134023212

[14] Dean J. and Ghemawat S. (2004) , MapReduce: simplified data processing on large clusters, Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation, volumen 6, pp 1-10.

[15] Apache Hadoop (Página web oficial): https://hadoop.apache.org/

[16] Apache Spark (Página web oficial): https://spark.apache.org/

[17] Tableau (Página web oficial): https://www.tableau.com

[18] Looker Business Intelligence (Página web oficial): https://looker.com/

[19] Microsoft Power BI (Página web oficial): https://powerbi.microsoft.com/es-es/