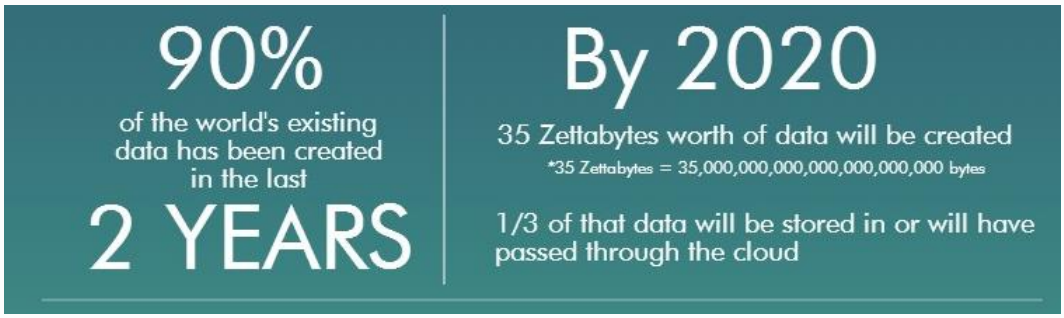
1. Infraestructura big data e introducción a analytics

1.1 Infraestructura big data

1.1.1 Conceptos fundamentales

¿Qué es *big data*? Esa es una de las primeras preguntas que surgen cuando hablamos de este concepto que es nuevo y está cada vez más presente en las empresas, en su relación con sus clientes, como así también en sectores que no tienen como principal objetivo el aumento de sus ingresos económicos, como el caso de las organizaciones de salud. *Big data* son datos que exceden la capacidad de procesamiento convencional de una base de datos, son datos de gran volumen y que se transfieren a gran velocidad o que tienen determinada estructura que no se adapta a las arquitecturas convencionales de base de datos. Es por eso que es necesario una nueva alternativa para procesar estos datos.

Figura 1: Datos



Fuente: [Imagen sin título, sobre volumen de datos], s.f., https://technophenia.files.wordpress.com/2015/07/dataoutlookpinfographjpeg.jpg

Desde no hace mucho tiempo el *big data* se transformó en una alternativa con una relación costo-beneficio aceptable para pequeñas y medianas empresas, que permite trabajar con grandes volúmenes de datos sin requerir grandes y costosas infraestructuras. Años atrás solo unas pocas empresas líderes a nivel mundial podían hacerlo, tales como Wal-Mart o Google. Pero el *commodity* de *hardware* actual, las arquitecturas *cloud* y el *software open source* permiten reducir los costos e implementar este tipo de

arquitecturas, incluso en empresas del estilo *startups* que nacen en un garaje.

El valor del *big data* en una organización está dada en dos categorías:

- Uso analítico: *big data analytics* permite encontrar *insights* ocultos en los datos que antes eran muy costosos de descubrir, por ejemplo, la influencia de la relación entre clientes que se puede hallar al analizar las transacciones comerciales, con datos sociales y geográficos. Brinda la posibilidad de procesar grandes volúmenes de información sin necesidad de realizar muestreos que pueden llevar a una solución menos precisa.
- Permite nuevos productos: *big data* brinda la posibilidad de crear nuevos productos y servicios con el uso de los datos. Por ejemplo, al combinar una gran cantidad de señales de las acciones de un usuario y de sus amigos, Facebook pudo lograr personalizar la experiencia del usuario y creó una nueva forma de negocios publicitarios.

¿Cómo luce el *big data*? Esta puede ser una terminología un poco confusa que no permite comprender exactamente que está dentro del concepto y qué no. Un dato de entrada en un sistema de *big data* puede ser todo lo relacionado a redes sociales, *logs* de una web, tráfico de distintos sensores, imagen satelital, audio, transacciones bancarias, seguimiento por GPS, y la lista puede continuar.

Para clarificar este concepto siempre se habla de las tres **V**: volumen, velocidad y variedad, que permiten caracterizar los diferentes aspectos de *big data*. En algunos casos se agrega una cuarta **V** que es "veracidad" como en el caso de IBM, que lo toma como algo necesario para unir las tres V anteriores, a modo recordatorio de la materia de "*Introducción a big data*" se describe a continuación cada una de ellas de forma resumida:

- **Volumen:** el beneficio de tener la habilidad de procesar grandes volúmenes de datos es el principal atractivo de *big data analytics*. Debido a que tener mayor cantidad de datos lleva a que puedan ser mejores modelos, un ejemplo de esto puede ser un modelo predictivo para hacer *forecasting* de las ventas en un mes determinado.

- **Velocidad**: la importancia de la velocidad de transferencia de los datos sigue el mismo patrón que el volumen. En la era de Internet y de los celulares el consumo de productos y servicios genera un flujo de datos al proveedor que hace necesario trabajar, en muchos casos, en forma *real-time*, es decir, en cada acción del cliente provoque una reacción instantánea y tenga algún resultado inmediato. Por ejemplo, alguna campaña o publicidad de lo que usted está consumiendo le propone lo que le puede interesar consumir después.

- **Variedad**: muy rara vez el dato se presenta de forma perfectamente ordenada y listo para ser procesado. Algo común en *big data* es que las fuentes de datos son diversas y no necesariamente tienen una estructura relacional. Pueden ser textos de redes sociales, imágenes, etc. Por lo tanto, *big data,* en estos casos, recibe estos datos en un formato no estructurados y brinda como resultado datos con un significado ordenado. Esto lleva a un principio de *big data*: *when you can, keep everything*.

Figura 2: Datos por día

Fuente: [Imagen sin título, sobre volumen de datos por día], s.f., https://technophenia.files.wordpress.com/2015/07/dataoutlookpinfographjpeg.jpg

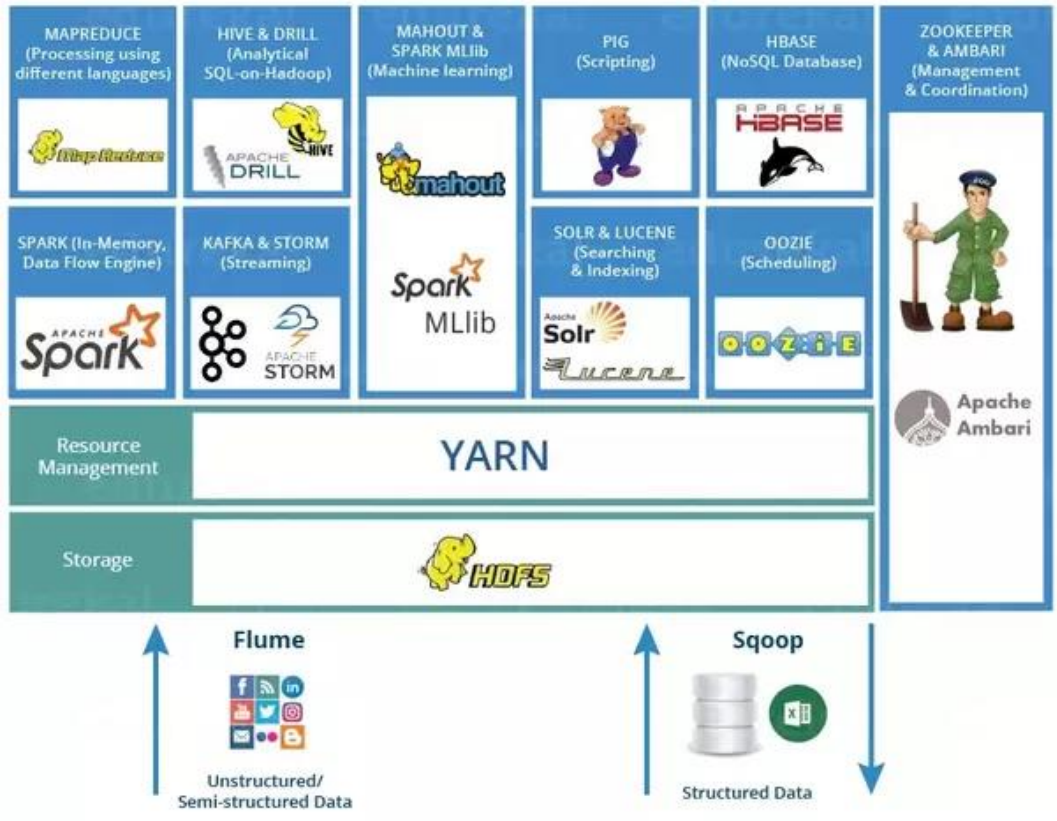
1.1.2 Ecosistema Hadoop

Hadoop es una plataforma distribuida que reparte los problemas computacionales entre los distintos servidores. Fue creado y desarrollado como *open source* por Yahoo!, y luego se agregaron las implementaciones de MapReduce liberadas por Google que optimizó la búsqueda indexada. El MapReduce de Hadoop distribuye los conjuntos de datos (*dataset*) entre múltiples servidores, los cuales operan de forma individual sobre los datos (esta etapa se llama *map*). Luego se combina el resultado parcial de cada servidor y se lo conoce con el nombre de *reduce*.

Este *framework* de *software* es una solución de *big data* que, con la combinación de todos sus elementos, se llama **ecosistema** y cumple con las 3 V explicadas en la sección anterior. Hadoop es visto como un conjunto de productos que forman el ecosistema de un zoológico, por eso mismo

cada uno de ellos representan algún animal o sector del mismo, y una tarea específica para responder a las necesidades de *big data*, que van a ser descriptas más adelante.

Figura 3: Ecosistema Hadoop



Fuente: [Imagen sin título, sobre Ecosistema Hadoop], s.f., https://qph.fs.quoracdn.net/main-qimg-703f5d1f767006583c71fb20da33cb10

Hoy en día la arquitectura es usada por la mayoría de las empresas multinacionales a lo largo del mundo y existen distintas versiones a partir del estándar, más el agregado de determinadas características por cada uno de los vendors, en lo que se pueden encontrar Cloudera, Hortonworks o MapR entre los más utilizados.

Algunos conceptos claves que describen las características de Hadoop son las siguientes:

- **Basado en sistemas distribuidos:** consiste en utilizar más de una maquina (servidores) para el procesamiento de un conjunto de tareas, pero es una arquitectura compleja y es por eso que Hadoop se basa en

este estilo de sistemas, pero con algunas mejoras aplicadas.

- **Distribución y procesamiento de los datos en los distintos servidores**: cada uno de los servidores contiene una porción de los datos, y los datos son procesados de forma individual en cada uno de estos. De esta forma evita inconvenientes de transferencia y demoras de procesamiento.

- **Comunicación entre nodos (servidores) lo menor posible**: los desarrolladores no tienen que escribir código para comunicar entre nodos y es totalmente transparente, tiene una arquitectura *shared nothing*.

- **Los datos que se cargan en el sistema son divididos**: los datos almacenados son divididos en bloques que varían entre 64 MB o 128 MB.

- **Existe un programa que administra al resto**: en la arquitectura existe un *master* que se dedica de alocar el trabajo en cada uno de los nodos para procesar sobre los bloques de datos. Si alguno de los nodos falla el master lo detecta y deriva el trabajo a otro nodo.

Todos estos conceptos se encuentran definidos y distribuidos por los componentes que forman parte de este ecosistema Hadoop. Existen dos elementos básicos, *HDFS* y *Map Reduce,* y otros componentes basados en proyectos alrededor del core de Hadoop, por ejemplo, *Hive, Pig, Sqoop,* etc.

HDFS

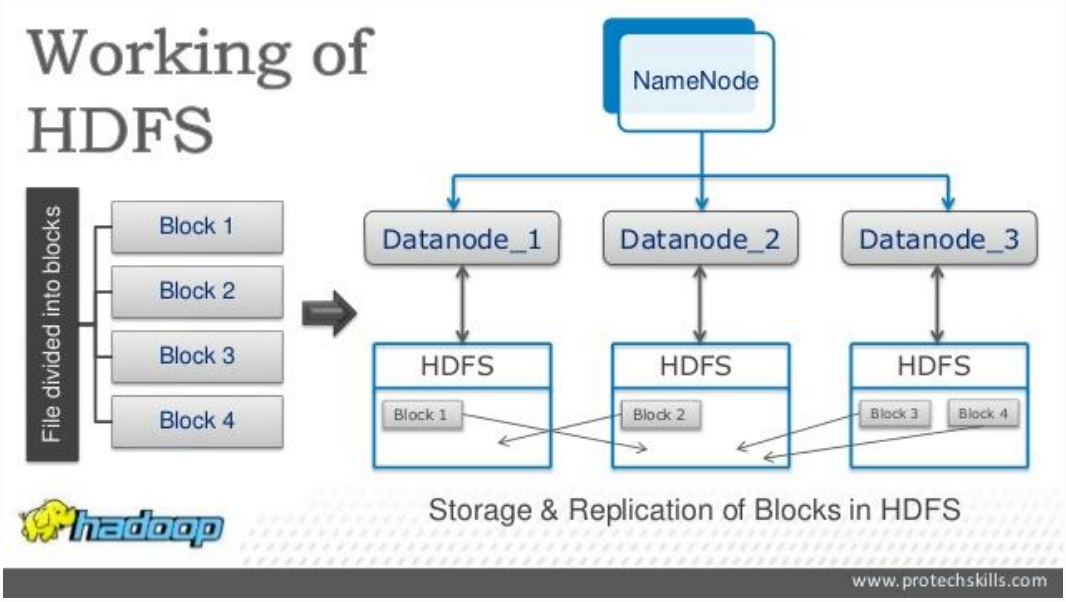
Toda la información es almacenada en un formato de datos especial dentro de Hadoop que recibe el nombre de Hadoop Distributed File System (HDFS), y se encuentra dentro del cluster de servidores (conjunto de servidores distribuidos).

HDFS se asegura que la información esté duplicada y que sea redundante a través del *cluster* y que, en el caso de que *Map Reduce* falle, pueda ser utilizado en cualquiera de los servidores que forman parte del *cluster*. Es decir, que los datos se reparten en bloques y se distribuyen entre los distintos nodos del cluster, y cada uno de estos bloques se multiplican una cierta cantidad de veces, generalmente 3 veces cada bloque en particular. De esta forma se asegura la confiabilidad y disponibilidad. El tamaño de cada uno de estos bloques es grande, a diferencia de otros File System como Linux que utiliza bloques de 4 KB, HDFS tiene bloques que van de 64 MB a 128 MB, debido a que al trabajar con grandes volúmenes de datos cada uno de estos bloques es seguido por una metadata, y al hablar de

Petabytes (1024 Terabytes) estaría bajo un inconveniente de overhead de metadata enorme.

La detección de fallas y la recuperación automática y rápida es un objetivo clave dentro de la arquitectura HDFS, es por eso mismo que un bloque se repite n veces en los distintos nodos del cluster.

Figura 4: Arquitectura HDFS



Fuente: [Imagen sin título, sobre Arquitectura HDFS], s.f., https://image.slidesharecdn.com/hadoophdfs-140826093616-phpapp02/95/hadoop-hdfs-concepts-4-638.jpg?cb=1447335462

Como se observa en la Figura 4 se encuentran los siguientes elementos y la función de cada uno:

- **File o Archivo**: es el archivo que se almacena en formato HDFS, puede ser de una gran cantidad de formato, como por ejemplo en formato de texto plano. Existen otros formatos de archivos que son efectivos para el uso en HDFS y para el resto de los componentes como *Hive* o *Impala*, entre los formatos más utilizados se encuentran: *Avro y Parquet*, que consiste en un almacenamiento columnar de los datos.

- **Block o Bloque**: cada uno de esos archivos es dividido en bloques de tamaños grandes para luego ser almacenados en los distintos nodos.

- **Name Node:** cuando un usuario almacena un archivo en HDFS, el archivo se divide en bloques y estos se copian tres veces más para luego

almacenar en cada data node. Este nodo sabe que datos contiene cada bloque y también donde está ubicado cada uno de ellos dentro del cluster.

- **Data Node**: también se los denomina Slave Node, cada uno de estos tienen información de las porciones de datos que el sistema almacena en ese nodo. Se comunica regularmente con el Name Node o Master para reportar la salud y estado de los datos almacenados localmente. Los Data Node no saben que tienen dentro de los bloques, pero si el Master que está a cargo de mapear cuales bloques de datos están almacenados en el HDFS.

La arquitectura HDFS implementa un modelo de datos: *Write Once, Read Often*. Por lo tanto, existen dos tipos de acciones que se pueden realizar con HDFS:

- **Escritura de datos**
    1. El cliente envía una acción de crear un nuevo archivo: el name node determina cuantos bloques va a necesitar y le da al cliente el permiso de crear estos nuevos bloques en el cluster.

    2. El cliente escribe las primeras copias de los bloques a los data nodes. El name node administra la escritura determinando donde los bloques y sus réplicas se van a escribir, balanceando disponibilidad y performance. Esto minimiza tráfico de red, asegurando que no existan bloques de datos en el mismo punto de falla.

    3. Mientras cada bloque se escribe en HDFS, un proceso especial escribe las réplicas en los data node especificados por el name node.

    4. Por último, el cliente informa que se creó el archivo y se libera el permiso.

- **Lectura de datos**
    1. El cliente envía un requerimiento de lectura de un archivo al name node. Para esto el name node determina cuales son los bloques involucrados y elige cual es el acceso más eficiente.

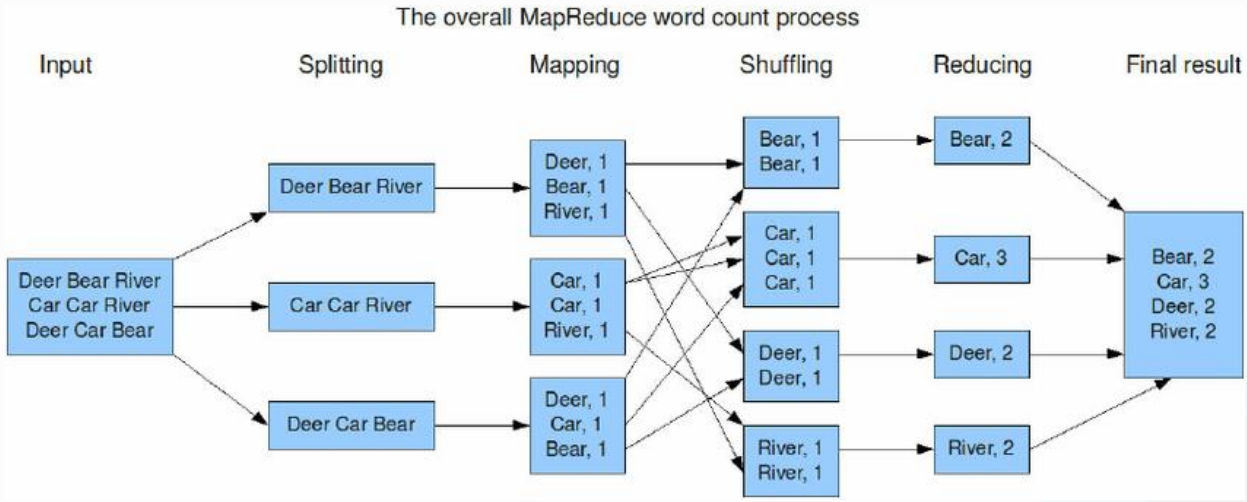
    2. El cliente accede a los bloques a partir de las direcciones dadas por el name node

Map Reduce

La innovación importante de este producto es la habilidad de realizar consultas sobre un *dataset*, dividirlo, y ejecutar en forma paralela sobre múltiples servidores. Estos servidores distribuidos solucionan el inconveniente de trabajar con grandes volúmenes de datos al permitir distribuirlos en volúmenes más sencillos de controlar en distintos servidores que trabajan de forma paralela. Por lo tanto, esta paralelización y distribución automática de las tareas en los distintos nodos se llevan a cabo por dos acciones:

- **Map:** Hadoop intenta asegurar que cada Mapper corre en nodos que mantiene su porción de datos localmente para evitar tráfico en la red, es decir, que múltiples mappers corren en paralelo, y cada uno procesa una parte de los datos de entrada. Por ejemplo, si se desea contar la cantidad de palabras repetidas en un archivo, cada uno de los nodos va a contar la cantidad de palabras en los bloques del archivo que se encuentra almacenado en su HDFS.
- **Reduce:** luego de que el map finaliza su tarea, todos los valores intermedios son combinados en una lista y el reducer se encarga de realizar de reducir la lista de todos los mappers y consolidar la misma. Continuando el ejemplo anterior, cada una de esas palabras que fueron contadas es listada en un sector en común para que el reducer genere la cuenta final de cada una de esas palabras.

Figura 5: Map Reduce

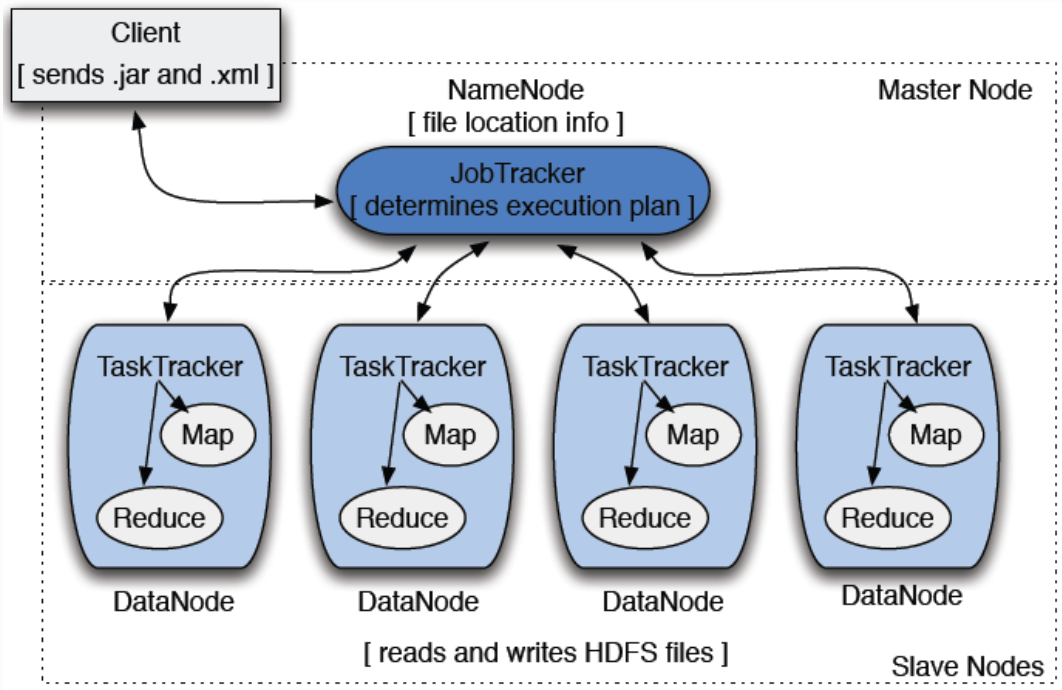


Fuente: [Imagen sin título, sobre Map Reduce], s.f., https://www.researchgate.net/profile/Oscar_Pereira3/publication/270448794/figure/fig6/AS:295098651824130@1447368409317/Word-count-program-flow-executed-with-MapReduce-5.png

Al igual que HDFS, Map Reduce se caracteriza por tener una arquitectura distribuida en donde cada uno de los nodos tiene una tarea específica de acuerdo al rol que desempeña, en los que podemos encontrar:

- **JobTracker:** es el nodo que se encarga de administrar las tareas de Map Reduce y las distribuye en los distintos nodos. Este JobTracker necesita ejecutarse sobre el Master Node del cluster Hadoop ya que coordina las tareas al resto de los nodos.
- **TaskTracker:** está asociado a cada data node y es el responsable de instanciar y monitorear las tareas Map y Reduce individuales.

Figura 6: Arquitectura Map Reduce



Fuente: [Imagen sin título, sobre Map Reduce], s.f., https://t1.daumcdn.net/cfile/tistory/1762633B4F571A051B

HIVE

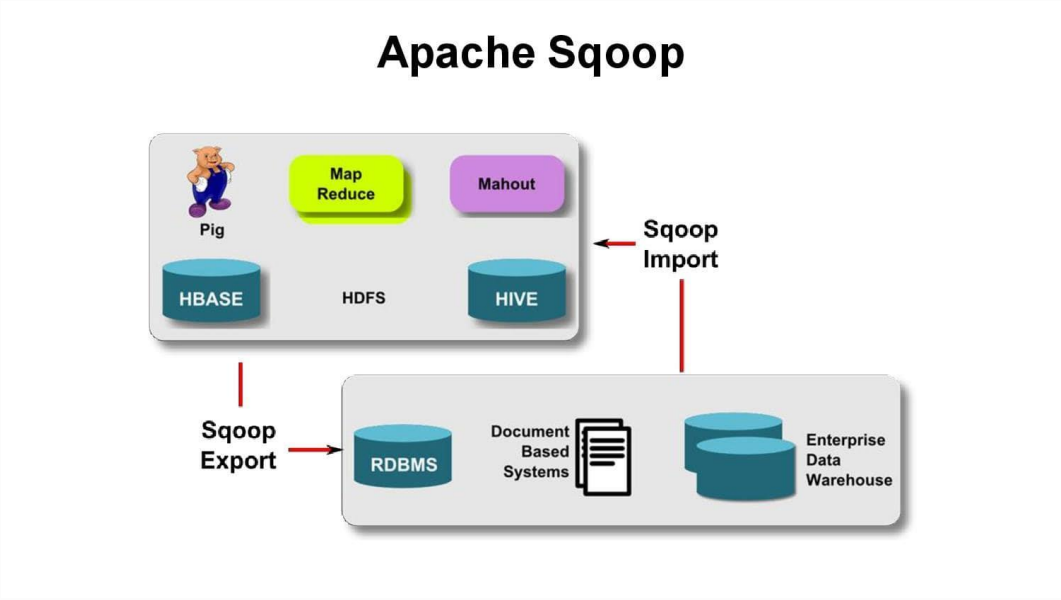
Permite a Hadoop operar con los datos como si fuera un *warehouse*. Al estilo de la sintaxis de consultas SQL, pero accediendo al HDFS en datos no estructurados. Por lo tanto, puede ser una buena opción para realizar consultas y análisis al estilo *data warehousing*, que pueden llevar a realizar tareas como sumarizaciones, queries ad-hoc o análisis de grandes volúmenes de datos.

SQOOP

Para poder transferir datos de una base de datos relacional a HDFS, Hadoop cuenta con Sqoop que crea copias de tablas o datos para almacenar en el *framework* y luego poder hacer consultas con *Hive* sobre los mismos. Es decir, que es factible copiar base de datos o tablas como de Oracle, SqlServer, MySql, etc, para luego dejarla almacenada en formato HDFS en el Hadoop y disponible para que elementos como *Hive* por medio de consultas HiveQL (SQL). Logrando de esta forma cumplir con la posibilidad de tener una gran variedad de tipos de datos.

Al mismo tiempo, es factible realizar extracción de datos en HDFS a una base de datos relacional, es decir, que se permite enviar datos a una base de datos que no soporte datos en el formato de Hadoop y de esta forma integrarse con otras plataformas que no forman parte del mundo *big data*.

Figura 7: Arquitectura Sqoop



Fuente: [Imagen sin título, sobre Arquitectura Sqoop], s.f., https://s3.amazonaws.com/files.dezyre.com/hadoop_page1.0/slides/flume/Sqoop_Flume_Oozie_slides-page-009-min.jpg

Zookeeper

Al ser tantos los servicios que se encuentran en ejecución y que forman parte del *cluster* de Hadoop, hay una necesidad de coordinarlos. Como los servidores pueden ir y venir, los miembros del *cluster* necesitan sincronizarse entre sí y requerir un conocimiento de donde acceder a los servicios y saber cómo deberían estar configurados. Surge así un nuevo producto que cumple con todas estas tareas y es Zookeeper.

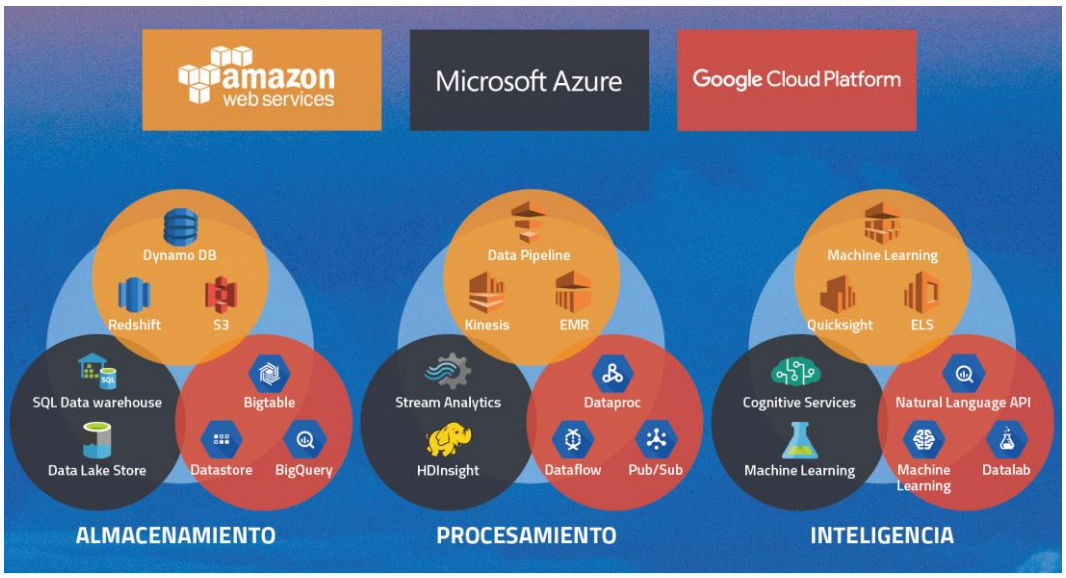
1.1.3 Plataformas y APIs en la nube

Para poder transformar los datos en conocimiento por medio de *big data* es necesario contar con una arquitectura que atienda nuestros requerimientos. Actualmente existen diversas opciones que permiten contar con estas herramientas y reducir los costos para aquellas empresas que no están aptas para afrontar.

Dentro de las opciones, las que más se destacan son:

- Amazon.
- Microsoft Azure.
- Google Cloud Platform.

Figura 8: ***Big data cloud***



Fuente: [Imagen sin título, sobre big data cloud, s.f., https://www.paradigmadigital.com/wp-content/uploads/2016/11/bigdataenlasnubes_Infografia_Infografia.png.

1.2 Introducción a analytics

1.2.1 Conceptos fundamentales

Big data permitió que grandes volúmenes de datos de distintos orígenes fueran almacenados a una velocidad de transferencia alta sin ningún tipo de restricción respecto de la variedad de los datos. Pero todos estos datos almacenados de distintas fuentes de información no se pueden valorizar si sólo se mantienen en estos repositorios, es por eso que estos datos se deben analizar y buscar responder inquietudes que surgen en el contexto actual y las que pueden surgir en el futuro.

Por lo tanto, una vez almacenados, es necesario realizar *analytics* de los datos, que consiste en buscar patrones de comportamiento para comprender mejor el dato en cuestión por medio de la combinación de distintas fuentes de información. Es decir, se realiza la transición de un dato, sin manipular, a ser una información procesada y finalizar como un conocimiento, en base a la analítica aplicada sobre el dato inicial.

Estas nuevas oportunidades conducen a un mejor conocimiento de la organización y saber qué acciones tomar para mejorar la relación del cliente con nuestros productos y servicios, de una forma más eficiente y con mejores resultados que los métodos tradicionales.

¿Cómo encontrar los patrones de mis datos? *Analytics* permite descubrir patrones de los distintos datos que forman parte de nuestro *big data*, pero para hacerlo existen diversas formas, según el enfoque que se desee hacer o según el problema que se intenta solucionar. Según sea el modelo matemático que utilicemos podemos hablar de tres enfoques para la detección de esos patrones: *machine learning, inteligencia artificial* y *deep learning*.

1.2.2 *Machine learning*

A pesar que *machine learning* existe hace décadas, diversos factores condujeron a su auge en estos últimos años, debido a tres factores:

- Volúmenes de datos cada vez mayores.
- *Hardware* más rápido y económico.
- Necesidad de contar con una diferencia competitiva.

¿Qué es *machine learning*? Es una disciplina científica que, por medio de algoritmos matemáticos, busca realizar un aprendizaje de los datos y lograr una predicción lo más precisa posible.

Consiste en utilizar resultados conocidos para entrenar un modelo que sirva para predecir valores para datos diferentes. Es decir, en base a datos del pasado, predecir nuevos valores. Por ejemplo, si deseo saber las ventas de una empresa en el verano, debería entrenar un modelo que integre los datos de las ventas de los veranos anteriores para poder predecir las del próximo verano, el resultado de esto me arrojaría una cierta probabilidad de que esto ocurra.

Existen dos tipos de modelos predictivos:

- Modelos de clasificación: consisten en predecir la membresía de una clase, es decir a qué clase corresponde cada dato. Por lo tanto, se desea clasificar en un número finito de clases. Por ejemplo, se puede utilizar para determinar si un correo electrónico es correo deseado o no y de esta forma eliminar todo correo que sea spam. Esta clase de ejemplo sólo tiene dos resultados posibles y sería una clasificación binaria.

- Modelos de regresión: es útil para predecir datos que son continuos, es decir, que la respuesta a una pregunta se representa en base a una cantidad y no a un valor finito que puede ser etiquetado como es el caso de la clasificación. Un ejemplo muy común es predecir la cantidad de ventas que se realizarán en un tiempo determinado para una empresa en particular, el resultado de esta respuesta es un valor numérico que identifica una cantidad, por lo tanto, estamos bajo la presencia de datos continuos.

1.2.3 Inteligencia artificial (AI)

Desde la década del ‘50 comenzó a utilizarse esta terminología para hacer referencia a la simulación de los comportamientos humanos por medio de máquinas. Esto puede incluir patrones diseñados para alguna tarea en particular o para tareas que requieren habilidades cognitivas y que permiten dar respuesta a escenarios desconocidos.

Un ejemplo claro de AI diseñados para una tarea particular es Siri de Apple. Esta herramienta fue desarrollada para brindar un servicio de asistencia personal virtual, para lo cual las combinaciones de todas las consultas previas ya están aprendidas por el algoritmo.

Un ejemplo de AI con habilidades cognitivas son los vehículos autónomos que tienen la capacidad de conducir en la ruta y reconocer los distintos obstáculos que se les presenta. Un buen resultado de esto es Olli, el vehículo autónomo que se utiliza de prueba como medio de transporte y que, al

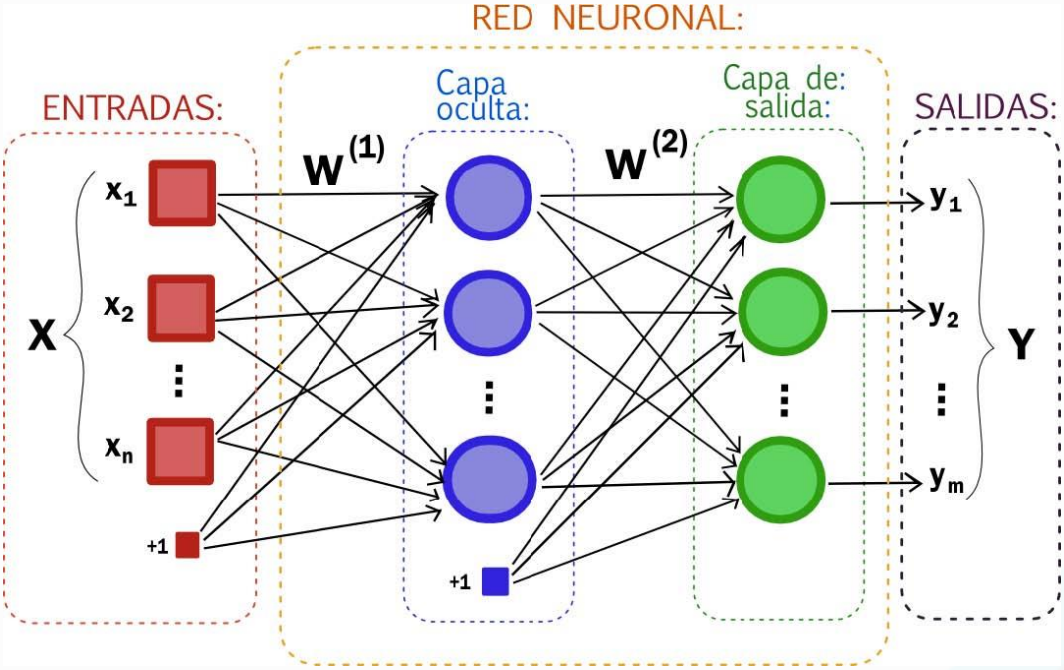
mismo tiempo, tiene la capacidad de ser un asistente virtual en el mismo viaje. Lo invitamos a ver el siguiente video para más detalles: https://www.youtube.com/watch?v=4XDwxvCQozg

1.2.4 *Deep learning*

Proviene de algoritmos de la clase de *machine learning* e imitan de forma artificial el cerebro humano: imitan las mismas funciones de las neuronas originales, pero a través de modelos matemáticos. Esta clase de modelos se los denomina **redes neuronales**, que utiliza algoritmos matemáticos que se caracteriza por tener una arquitectura por capas que, por lo general, se dividen en tres:

- De entrada: son aquellas neuronas que reciben la información.
- Ocultas: son las encargadas de realizar los cálculos intermedios de la red.
- De salida: son las neuronas que contienen el resultado del modelo.

Figura 9: Red neuronal



Fuente: [Imagen sin título, sobre redes neuronales], s.f., https://sites.google.com/site/mayinteligenciartificial/_/rsrc/1470337417111/unidad-4-redes-neuronales/3.jpg.

Cada una de las neuronas tienen un número de importancia en la red y, al mismo tiempo, todas están conectadas entre sí con un peso de importancia de la conexión entre ellas.

Actualmente el *deep learning* tiene mucha fuerza en varios campos de las organizaciones, pero aún si es un concepto muy reciente en cuanto a su uso. Algunas de las aplicaciones actuales son para el reconocimiento de caras, voz, como así también para interpretación semántica.

Referencias

[Imagen sin título, sobre volumen de datos] *(s.f.) Recuperada de https://technophenia.files.wordpress.com/2015/07/dataoutlookpinfographjpeg.jpg*

[Imagen sin título, sobre volumen de datos por día] *(s.f.) Recuperada de https://technophenia.files.wordpress.com/2015/07/dataoutlookpinfographjpeg.jpg*

[Imagen sin título, sobre Ecosistema Hadoop] *(s.f.) Recuperada de https://qph.fs.quoracdn.net/main-qimg-703f5d1f767006583c71fb20da33cb10*

[Imagen sin título, sobre Arquitectura HDFS] *(s.f.) Recuperada de https://image.slidesharecdn.com/hadoophdfs-140826093616-phpapp02/95/hadoop-hdfs-concepts-4-638.jpg?cb=1447335462*

[Imagen sin título, sobre Map Reduce] *(s.f.) Recuperada de https://www.researchgate.net/profile/Oscar_Pereira3/publication/270448794/figure/fig6/AS:295098651824130@1447368409317/Word-count-program-flow-executed-with-MapReduce-5.png*

[Imagen sin título, sobre Map Reduce] *(s.f.) Recuperada de https://t1.daumcdn.net/cfile/tistory/1762633B4F571A051B*

[Imagen sin título, sobre Arquitectura Sqoop] *(s.f.) Recuperada de https://s3.amazonaws.com/files.dezyre.com/hadoop_page1.0/slides/flume/Sqoop_Flume_Oozie_slides-page-009-min.jpg*

[Imagen sin título, sobre big data cloud] *(s.f.) Recuperada de https://www.paradigmadigital.com/wp-content/uploads/2016/11/bigdataenlasnubes_Infografia_Infografia.png*

[Imagen sin título, sobre redes neuronales] *(s.f.) Recuperada de https://sites.google.com/site/mayinteligenciartificial/_/rsrc/1470337417111/unidad-4-redes-neuronales/3.jpg*

Sturges, D. *(9 de diciembre de 2016). James Corden goes for a ride in OLLI - Autonomous Shuttle by Local Motors [YouTube]. Recuperado de https://www.youtube.com/watch?v=4XDwxvCQozg*