

# Diseño de una arquitectura para Big Data

Esteban Sevilla Rivera y David Salas Vargas

Escuela de Ingeniería,  
Universidad Latinoamericana de Ciencia y Tecnología,  
ULACIT, Urbanización Tournón, 10235-1000  
San José, Costa Rica  
<http://www.ulacit.ac.cr>

**Abstract.** La información es un recurso muy importante que permite generar conocimiento utilizado por diferentes sectores de la industria para crear innovación. Actualmente, esta información está creciendo cada vez más por el uso de diferentes orígenes de datos provenientes de Internet como las redes sociales, lo que da como resultado un problema a la hora de procesarla para obtener el conocimiento requerido. Este trabajo toma en consideración una revisión detallada de las arquitecturas que han sido propuestas por los principales actores en la industria, así como de artículos científicos en los cuales se definen elementos de relevancia. Con base en dicha revisión, se determinan los principales elementos de cada arquitectura, se efectúa una comparación de estas y se define un modelo genérico de arquitectura que toma en cuenta las principales fortalezas de las arquitecturas que han sido estudiadas. Posteriormente, con base en el modelo de arquitectura, se definen dos casos de uso en los cuales se proponen soluciones para problemas de análisis de datos. Lo anterior, tomando en consideración las posibles herramientas que pueden ser utilizadas por la arquitectura.

**Keywords:** Big Data, Big Data Analytics, Arquitectura Big Data.

## 1 Introducción

En la actualidad se generan grandes volúmenes de datos de diversos tipos, a gran velocidad y con diferentes frecuencias. Las tecnologías disponibles permiten efectuar su almacenamiento, adquisición, procesamiento y análisis utilizando métodos y técnicas diversas. Es importante tener en cuenta que cuando se procesan y almacenan grandes volúmenes de datos entran en juego dimensiones adicionales a las técnicas, como el gobierno, la seguridad y las políticas. Por lo que elegir una arquitectura y desarrollar una solución de Big Data es un proceso complejo que requiere considerar muchos factores de acuerdo con cada problema particular. En este documento se relacionan diferentes aspectos sobre los patrones de Big Data con el fin de presentar un enfoque estructurado para simplificar la tarea de definir una arquitectura genérica para el procesamiento y análisis de Big Data. Debido a la importancia evaluar si un escenario empresarial constituye un problema de Big Data, se incluyen indicios para determinar qué

problemas empresariales son buenos candidatos para aplicar soluciones de Big Data.

## 2 Objetivo general

Proponer una arquitectura para Big Data Analytics a partir del estudio de las propuestas de la industria y la academia.

### 2.1 Objetivos específicos

1. Identificar las tecnologías, métodos y técnicas utilizadas por Big Data para el almacenamiento, análisis avanzado de datos y la presentación de los resultados.
2. Comparar algunas arquitecturas que se han efectuado a nivel de la industria y en las propuestas académicas.
3. Definir una arquitectura para Big Data Analytics y casos de uso de dicha arquitectura (ejemplos concretos de aplicación).

## 3 Big Data

Durante los últimos años, la manera en que las personas acceden los datos y sus orígenes han cambiado drásticamente. Lo anterior, como resultado del avance tecnológico. Ahora bien, esta revolución tecnológica genera grandes cantidades de datos que provienen de diferentes orígenes, llámese estas redes sociales, sistemas tecnológicos o paginas web utilizadas, para marketing como Amazon. Por esta razón, Big Data es un concepto que hace referencia a grandes cantidades de información disponibles en diferentes formatos y tipos de estructuras recopiladas principalmente a través de Internet mediante la interacción de usuarios de computadores, teléfonos móviles, dispositivos GPS entre otros.

### 3.1 Las cinco Vs

Cuando se habla de Big Data, se hace referencia a la producción de grandes cantidades de datos que provenientes de diferentes orígenes de datos. Sin embargo, Big Data representa más que el concepto de grandes cantidades de datos y por eso se define el modelo de las Vs que comenzó con el modelo de las tres Vs propuesto por IBM y luego mejorado al agregar dos V mas, las cuales dependen del objetivo que se quiera lograr mediante este sistema.

A continuación se describe brevemente cada una de las Vs:

Este modelo de las cinco Vs permite determinar las características principales que debe tener una propuesta de arquitectura para Big Data, ya que tiene que cumplir con estos principios de procesamiento en masa, obtención de diferentes formatos, velocidad de generación y adaptación a cambios.

V de Big Data	Concepto
<b>Volumen</b>	Se refiere a la enorme cantidad de datos (zettabytes y brontobytes) que son generados cada segundo y que provienen de orígenes de datos como las redes sociales y los sistemas inteligentes.
<b>Velocidad</b>	Se refiere a la velocidad con que los datos son generados y se mueven a través de internet de las cosas. La tecnología actualmente permite analizar esta información mientras es generada y muchas veces procesada en memoria sin la necesidad de almacenarla en una base de datos.
<b>Variedad</b>	Se refiere a los diferentes tipos de datos que se pueden usar. En el pasado se trabajaba principalmente con datos estructurados como datos financieros y que podían ser almacenados en bases de datos relacionales. Por otro lado, actualmente se habla que el 80% de los datos en el mundo son no-estructurados (imágenes, videos, voz, etc.) por lo que Big Data encaja en este punto ya que permite procesar datos de diferentes tipos y provenientes de diferentes orígenes.
<b>Variabilidad</b>	Se refiere a la adaptación que Big Data es capaz de sostener por ejemplo a nuevos tipos y orígenes de datos así como su procesamiento que son resultado de la evolución constante de la tecnología.
<b>Valor</b>	Se refiere a la generación de valor o conocimiento a través del procesamiento de Big Data y que pueden dar como resultado la generación de innovación para las compañías.

Fig. 1. Las cinco Vs de Big Data.

### 3.2 Tipos de Datos

Como bien se menciona en el modelo de las Vs, la generación de datos proviene de diferentes orígenes y tipos de datos que necesitan ser transformados, procesados, almacenados y analizados. Por lo tanto, a continuación se describe cada uno de los tipos de datos que conforman la cola de información que dan como resultado la producción de Big Data:

Tipo de Dato	Descripción
<b>Estructurados</b>	Los datos ingresados tienen bien definido su longitud y formato (Ejemplos: Fechas, números, cadenas de caracteres, etc.). Se almacenan en tablas (Base de datos relacionales).
<b>No estructurados</b>	Poseen un formato tal y como fueron recolectados, los cuales carecen de un formato específico (Ejemplos: Archivos pdf, documentos multimedia, emails, etc.)
<b>Semiestructurados</b>	No se limitan a campos determinados, mantienen marcadores para separar elementos. Pueden contener información poco regular como para ser gestionada de una forma estándar (Ejemplos: Lenguajes de marca de hipertexto, lenguaje de marca extensible, etc.)

Fig. 2. Tipos de Datos.

Unas de las principales características de Big Data es la capacidad de soportar la variedad y variabilidad que conlleva la evolución tecnológica; asimismo, y que permite que estos datos puedan generar conocimiento valioso para los diferentes sectores de la industria.

Codigo	Nombre	Apellido	Edad
1	David	Salas Vargas	25
2	Esteban	Sevilla	30
3	Diego	Acosta	30

Fig. 3. Ejemplo de Tipo de Datos Estructurados.

```

<personas>
<persona>
  <nombre orden="primero">Karla</nombre>
  <nombre orden="segundo">Rosa</nombre>
  <apellido orden="primero">Rojas</apellido>
  <apellido orden="segundo">Segura</apellido>
  <nacionalidad>Italiana</nacionalidad>
</persona>
<persona>
  <nombre>Carlos</nombre>
  <apellido orden="primero">Juan</apellido>
  <apellido orden="segundo">Meza</apellido>
  <nacionalidad>Panameno</nacionalidad>
  <nacionalidad>Canadiense</nacionalidad>
</persona>
</personas>

```

Fig. 4. Ejemplo de Tipo de Datos semi-estructurados. Un fichero XML con información de personas. Los campos con la información de una persona están definidos pero pueden variar (la primera persona tiene dos nombres y la segundo dos nacionalidades).

### 3.3 Arquitectura de Big Data

En términos generales, una arquitectura de Big Data (Chunmei Duan1, 2014) esta compuesta por cinco componentes: recolección de datos, almacenamiento, procesamiento de datos, visualización y administración. Además, cada uno de estos componentes ha ido añadiendo nuevas tecnologías, las cuales dependen de las necesidades que se den y también su adaptación es mandatorio para dar una solución eficiente a las empresas actualmente. A continuación en la Figura 5,

se muestra el flujo general de una arquitectura de Big Data desde su origen de datos hasta su visualización y administración:

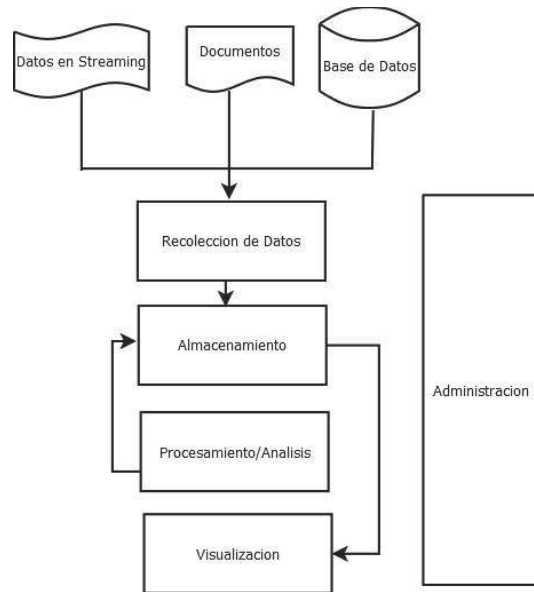


Fig. 5. Arquitectura General de Big Data.

### Recolección de datos

Esta etapa se refiere a la obtención de los datos. El sistema se conecta a las diferentes fuentes de información para obtener los datos que luego serán procesados, almacenados y posteriormente analizados.

A continuación, se describen los dos métodos de recolección de datos que se pueden dar y que dependen del caso de uso por desarrollar según el criterio:

- *Batch* o por lotes: Este tipo de recolección se conecta cada cierto tiempo a las fuentes de información, llámese estas: sistemas de ficheros o Base de Datos; en las cuales se buscan cambios realizados desde la última conexión que se hizo.
- *Streaming* o por transmisión en tiempo real: Este tipo de recolección trabaja directamente con la fuente de información de manera continua y de forma que la información se obtiene cada vez que se tramita (tiempo real).

Gracias a la evolución tecnológica, los sistemas de la actualidad pueden trabajar obteniendo la información de las dos formas usando *streaming* o *batch*

según la solución que se requiera. Asimismo, los sistemas modernos permiten filtrar la información; por ejemplo, según la información o formato que se quiere recolectar.

### **Almacenamiento**

Actualmente, los sistemas de almacenamiento han tenido que adaptarse o buscar nuevas formas de almacenar su información debido a las grandes cantidades de información que se generan además de la velocidad con que se mueven. Por esta razón, los métodos de almacenamiento tradicionales como las Bases de Datos relacionales se han quedado cortos a la hora de tratar esta información conocida como Big Data. Por lo tanto, se habla de sistemas de ficheros para el almacenamiento de Big Data los cuales presentan flexibilidad a la hora de almacenar información que contiene características como la variabilidad, velocidad de generación, volumen de datos, etc. “(Chunmei Duan1, 2014)”.

### **Procesamiento de datos**

Este paso presenta uno de los puntos más importantes a la hora de hablar sobre Big Data ya que una vez que se tienen almacenados los datos, se busca obtener conocimiento o valor por medio del procesamiento y análisis de toda esta información almacenada. Actualmente, se cuenta con herramientas muy poderosas que permiten procesar esta información que muchas veces presenta diferentes tipos de formato y orígenes como las bases de datos NoSQL o los sistemas de ficheros. “(Chunmei Duan1, 2014)”.

### **Visualización y Administración**

Esta capa de Big Data, muestra el producto del almacenamiento y procesamiento de la información que da como resultado la producción de conocimiento. Este conocimiento es de vital importancia para los diferentes sectores de la industria ya que presenta la oportunidad de innovación y desarrollo de nuevas líneas de negocios para sus organizaciones. “(Chunmei Duan1, 2014)”.

## **3.4 Arquitecturas existentes de la industria**

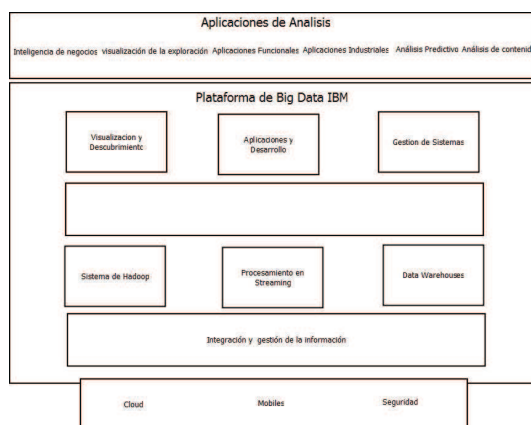
### **Arquitectura de IBM: BigInsights**

BigInsights es una plataforma de software para descubrir, analizar y visualizar datos de diferentes orígenes. Este software es normalmente utilizado para ayudar a procesar y analizar el volumen, variedad y velocidad de datos que entra continuamente a las diferentes organizaciones cada día. Asimismo, es una plataforma flexible construida sobre el Framework de código abierto Apache Hadoop que es ejecutado en paralelo sobre hardware de bajo costo. Esta solución de IBM ayuda a los desarrolladores, científicos de datos y administradores en las

organizaciones a rápidamente construir y desplegar análisis personalizados de información mediante el procesamiento de los datos. Estos datos son a menudo integrados en las bases de datos existentes, almacenes de datos (Data Warehouses) y la infraestructura de inteligencia de negocios. Además, mediante el uso de BigInsights, los usuarios pueden extraer nuevos conocimientos a partir de estos datos para mejorar el conocimiento de su negocio. “(IBM, n.d.)”.

### Características y arquitectura de BigInsights

Esta plataforma brinda un conjunto de herramientas o componentes tecnológicos que brindan las capacidades necesarias para que una organización pueda procesar los grandes volúmenes de datos recibidos diariamente. En la figura a continuación, se muestran los diferentes componentes que forman la arquitectura de BigInsights:



**Fig. 6.** Arquitectura de IBM BigInsights.

BigInsights extiende el Framework de Hadoop con seguridad a nivel empresarial, administración, disponibilidad, integración con almacenes de datos (Data Warehouses) existentes, además de herramientas que simplifican la productividad de los desarrolladores. Hadoop ayuda a las empresas a aprovechar los datos que antes era difícil de manejar y analizar. BigInsights cuenta con Hadoop y sus tecnologías relacionadas como elemento fundamental.

A continuación describimos algunos de los componentes utilizados en la arquitectura:

#### HDFS

HDFS viene con la plataforma abierta de IBM (IBM Open Platform) mediante el uso de Apache Hadoop como su sistema de archivos distribuido.

#### MapReduce

El Framework MapReduce es el núcleo de Apache Hadoop. Este paradigma de programación prevé escalabilidad masiva a través de cientos o miles de servidores en un clúster Hadoop.

### **Tecnologías de código abierto**

Hadoop incluye muchas tecnologías de código abierto y sus dependencias que continúan aumentando a medida que se utiliza Hadoop en más aplicaciones. Se emplean estas tecnologías para interactuar con el ecosistema de Hadoop.

#### **Text Analytics**

BigInsights incluye Text Analytics, que extrae información estructurada a partir de datos no estructurados y semi-estructurados.

#### **IBM BigSQL**

BigSQL aprovecha la fuerza de IBM en los motores de SQL para proporcionar acceso ANSI SQL a los datos a través de cualquier sistema de Hadoop, vía JDBC o ODBC y obteniendo los datos de Hadoop o una base de datos relacional.

#### **Integración con otros productos de IBM**

BigInsights complementa y amplía las capacidades de negocio existentes mediante la integración con otros productos de IBM. Estos puntos de integración extienden las tecnologías existentes para abarcar más tipos de información, lo que permite una visión completa de su negocio. Algunos productos de IBM con los cuales se puede trabajar a la hora de usar Hadoop tenemos los siguientes:

- IBM DB2.
- IBM Cognos Business Intelligence.
- InfoSphereStreams.
- IBM Netezza.
- IBM InfoSphere Data Explorer.

### **Arquitectura de Hortonworks**

Es una plataforma de código abierto “(Gorkahurtado, n.d.)” en la que todos sus componentes están desarrollados por medio de la Fundación Apache Software. Asimismo, proporciona todas las capacidades esenciales de Hadoop junto con una serie de características necesarias para el despliegue a nivel de empresa entre las que se encuentran la gestión de datos, gestión del acceso a datos, integración, gestión de recursos, seguridad y gestión de operaciones.

A continuación, se describe brevemente las diferentes funcionales integradas en Hortonworks:

#### **Gestión de Datos:**

Al ser una arquitectura basada totalmente en Hadoop, cuenta con HDFS el cual proporciona el almacenamiento escalable, distribuido y tolerante a fallos. Además, de poder utilizar YARN el cual proporciona la gestión de recursos del clúster lo cual permite el procesamiento de datos en paralelo.

#### **Acceso a Datos e Integración:**

Al utilizar YARN, Hortonwork permite el procesamiento de datos mediante el uso de varios de sus componentes que dan la oportunidad de interactuar con los datos de diferentes maneras, como por ejemplo el procesamiento batch usando MapReduce, NoSQL con HBASE y SQL con Hive. Además, por un lado, acepta el uso de lenguajes de *scripting* con *Pig* y el uso del procesamiento *streaming* con Apache Storm. Por otro lado, HDP proporciona herramientas como Falcon,



Oozie, Sqoop, o Flume que permiten obtener información de muchos orígenes de datos que pueden generar cualquier tipo de dato sean estos estructurados o no.

#### **Seguridad:**

Knox es la pieza central del sistema de autenticación. Este es uno de los componentes críticos que utiliza HDP para la autenticación, autorización y protección de datos.

#### **Gestión de Operaciones:**

Apache Zookeeper viene incorporado en la arquitectura de HDP y la cual brinda soporte para el despliegue, la monitorización y la gestión del clúster Hadoop lo cual brinda información estadística que dan visibilidad sobre la salud del sistema. Además, proporciona la capacidad para la configuración y optimización del rendimiento. Asimismo, el API Apache Ambari permite la integración con herramientas existentes de gestión de sistemas como Microsoft System Center o TeradataViewPoint.

#### **Arquitectura de Amazon**

Amazon es conocido por su plataforma en la nube, pero también ofrece un número de productos de Big Data, incluyendo el ElasticMapReduce basado en Hadoop, la base de datos Big Data DynamoDB, el almacén de datos paralelamente masivo RedShift, y todos funcionan bien con Amazon Web Services.

#### **Arquitectura de Google**

Las ofertas de Big Data de Google incluyen BigQuery, una plataforma de análisis de Big Data basada en la nube. Además Google ha sido uno de los principales precursores de Big Data ya que Hadoop está basado en los documentos originales de Google de MapReduce y de Google File System, el sistema de ficheros distribuido de Google.

#### **Arquitectura de Cloudera**

Cloudera “(Cloudera, n.d.)” está en la lista de los principales proveedores de Big Data con más de 141 millones de dólares en fondos de capital de riesgo y ha atraído a varios fundadores conocidos y de gran nombre en Big Data que vienen de Google, Facebook, Oracle y Yahoo. La compañía lanzó por primera vez la plataforma Apache Hadoop para clientes empresariales en el 2008. “(Gorkahurtado, n.d.)”.

Aspectos importantes:

- CDH no solo incluye el núcleo de Hadoop (HDFS, MapReduce. . .) sino que también integra diversos proyectos de Apache (HBase, Mahout, Pig, Hive, etc.)
- Cuenta con una interfaz gráfica propietaria, Cloudera Manager, para la administración y gestión de los nodos del clúster Hadoop.(Gorkahurtado, n.d.)
- La descarga es totalmente gratis, no obstante, también cuenta con una versión empresarial, que incluye una interfaz más sofisticada.

#### **Arquitectura de MapR**

Es mayormente conocido por su base de datos NoSQL M7, MapR funciona con la plataforma de Amazon en la nube y con Google Compute Engine. MapR reemplaza HDFS y en su lugar utiliza su propio sistema de archivo propietario,

llamado MapRFS. Este último ayuda a incorporar características de nivel empresarial en Hadoop, lo cual permite una gestión más eficiente de los datos, la fiabilidad y lo más importante, la facilidad de uso. “(Editor, n.d.)”.

Aspectos importantes de MapR:

- Hasta su edición M3, MapR es libre, pero, la versión gratuita carece de algunas de sus características propias (JobTracker HA, NameNode HA, NFS-HA, Mirroring, Snapshot entre otras).
- La edición M3 de MapR para Apache Hadoop se integrará en el sistema operativo Ubuntu por medio de una alianza reciente con Canonical (creador del sistema operativo Ubuntu).

### **Arquitectura de ORACLE**

Aunque Oracle es conocido principalmente por su base de datos, también ha incursionado muy bien en el ámbito de Big Data. Su Oracle Big Data Appliance combina un servidor Intel, distribución Hadoop de Cloudera y la base de datos NoSQL de Oracle. Además, Oracle combina una gran gama de herramientas del ecosistema de Hadoop para complementar muchas de sus capacidades.(Oracle, n.d.)

Aspectos importantes de la arquitectura de Oracle:

Oracle en su arquitectura propone 3 capas más que no se encuentran en ninguna de las otras arquitecturas:

- La capa de la infraestructura compartida que incluye el hardware y las plataformas en las que se ejecutan los componentes de Big Data: Infraestructura de bases de datos, Infraestructura de Big Data y la Infraestructura para análisis.
- La capa de servicios que incluye los componentes que prestan servicios de uso: Presentación de servicios, Servicios de información, Actividades de monitoreo del negocio, Reglas del Negocio y Manejo de Eventos. Los dos primeros están orientados a servicios de arquitectura (SOA). Los demás proporcionan servicios para la capa de procesamiento.
- La capa de distribución por múltiples canales, la cual los resultados pueden ser entregados por distintos medios ya sea por computadoras de escritorio, laptops, teléfonos celulares, tabletas, emails, entre otros.

### **Arquitectura de Spark**

Es un Framework Open source de procesamiento de datos en memoria, fue desarrollado en 2009 en UC Berkeley AMPLab. Es 100 veces más rápido que Hadoop. Ahora bien, MapReduce Soporta Java, Scala o Python, esto se debe a que el procesamiento lo hace en memoria RAM y no el disco; ya que este último es más lento. Asimismo, funciona implementando micro batches de datos llamados RDD (Resilient Distributed Dataset). Los anteriores, son una colección de elementos tolerantes a fallos que puedan ser procesados en forma paralela.

Datos importantes:

- Se crean RDD, se aplican transformaciones del RDD, se aplican acciones al RDD y se guardan los datos.
- El RDD es inmutable igual que la los datos en Hadoop.
- Una ventaja enorme que ofrece Spark es que permite extraer datos de Hadoop y mantenerlos en memoria, o mantener variables para su procesamiento y transformación.
- Spark extiende el modelo de computación de Map Reduce ofreciendo una riqueza en el lenguaje más amplia para resolver escenarios más complejos.

### 3.5 Tabla comparativa de las arquitecturas estudiadas

Categorías	Herramientas	Soluciones										
		Hadoop	Cloudera	Hortonworks	Google	Facebook	LinkedIn	IBM	ORACLE	Splunk	MapR	
Almacenamiento de Datos	Hadoop	x	x	x	x		x	x	x			x
	HDFS	x						x	x			x
	YARN							x	x			x
Procesamiento de Datos	Apache HBase	x					x	x	x			
	MapReduce / GFS				x							
	MapReduce							x	x			
	Spark Core											x
Acceso a datos	Apache Mahout		x						x			
	Apache Pig		x						x	x		
	Apache Hive		x					x	x	x		
	Sqoop		x									
	Cloudera Impala		x									
	Cloudera Search		x									
	DataFu		x									
	Dremel					x						
	Eventflow					x						
	MySQL Gateway					x						
	Bigtable					x						
	Scrite							x				
	SparkSQL											x
Administración	Hue		x									
	Oozie		x									
	Flume											
	Zookeeper		x					x				
	Chubby					x						
Aplicaciones ETL	Cloudera Manager		x									
	Snowall					x						
	Hipati							x				
	Spark Streaming											x
	Spark											x

Fig. 7. Tabla comparativa de las arquitecturas

### 3.6 ¿Cuál es el panorama actual de Big Data?

#### Nuevo panorama:

La inclusión de Big Data esta cambiando la manera en que las industrias mueven sus negocios en el mercado. Los limites que han surgido a lo largo del tiempo en los negocios han sido reemplazados por la utilización de nuevas tecnologías que se apoyan en la recolección y análisis de datos a gran escala.

#### Big Data cambiando el mundo de los negocios:

- Recortando gastos: El análisis de datos le permite a las empresas cumplir con los procesos y las normas establecidas. Además, este análisis se hace mediante el uso de herramientas gratuitas como Hadoop que representan un ahorro 20 veces menos por cada petabyte almacenado en relación con los métodos tradicionales de almacenamiento.
- Beneficios de Big Data: La mayoría de empresas presentan beneficios mediante el uso del análisis de Big Data; ya que ayudan a mejorar sus líneas de negocios.
- Nuevos mercados: La explotación de Big Data da como resultado la inclusión de nuevos competidores a los diferentes sectores de la industria además de la utilización de nuevas técnicas de marketing.

#### **Adoptar el cambio:**

Actualmente, es de vital importancia que las diferentes compañías adopten el cambio e inviertan en la infraestructura necesaria para el manejo de Big Data, para así, poder desarrollar nuevas técnicas y líneas de negocios que les permitan mantenerse y competir en el mercado cambiante.

## **4 Hadoop**

Es un Framework libre desarrollado por la Fundación Apache Software utilizado para el procesamiento distribuido de grandes cantidades de datos a través de clúster de computadoras y utilizando un modelo simple de programación llamado MapReduce “(Apache, n.d.)”.

### **4.1 Principales componentes:**

#### **HDFS(Sistema distribuido de archivos para Hadoop):**

Es un sistema que permite almacenar información desde un origen de datos hacia múltiples computadoras. Asimismo, opera bajo una arquitectura maestro-esclavo y esta compuesto por dos componentes principales:

NameNode el cual se encarga de:

- Manejar la información de los bloques de datos que hay en cada DataNode.
- Almacenar información acerca de cuantas veces un archivo se ha sido replicado en el clúster.
- Almacenar información acerca de cuantos bloques forman un archivo

Además, se encuentra el DataNode (Esclavo) el cual se encarga de:

- Procesamiento de datos.
- Almacenamiento de los bloques actuales.

**MapReduce:**

Es un framework de programación utilizado por Hadoop para la distribución de tareas en los nodos del clúster y para el procesamiento distribuido de datos. La base de MapReduce esta compuesta por dos funciones principales para el procesamiento de datos Map y Reduce. Un trabajo de MapReduce se encarga de de dividir los datos de entrada en un conjunto de bloques independientes que luego son procesados en paralelo por la función Map. Una vez terminada la función de Map, el framework ordena los resultados obtenidos por las funciones Map y luego las envía a ser procesadas por la funcion Reduce. Usualmente, las salidas y entradas de datos son almacenada por sistemas de archivos como por ejemplo HDFS. Además, el framework se encargar del monitoreo de las tareas programadas y la re-ejecución de las tareas fallidas.

## 5 Ecosistema de Big Data

El procesamiento de Big Data conlleva a la utilización de muchas de las tecnologías existentes en la actualidad. Además, involucra la elección de proyectos o herramientas que pueden ser *open source* o comercial. Por lo tanto, a continuación se presenta una breve descripción de alguna de las principales tecnologías de Big Data que hay en el mercado y que podrían presentar una buena solución para las compañías según la necesidad que se presente “(Hortonworks, n.d.)”.

### 5.1 Acceso a Datos

**Flume:**

Es un sistema distribuido para la recolección, agregación, y movilización de grandes cantidades de datos no estructurado y semi-estructurados desde múltiples fuentes en HDFS u otro almacén de datos central, principalmente datos en los cuales la generación de logs( WebSites, syslogs, STDOUT).

**Scribe:**

Es un servidor para la obtención de datos de logs que se transmiten en tiempo real. Está diseñado para ser escalable y soportar fallos. Está compuesto por un servidor Scribe que se ejecuta en cada uno los nodos del sistema, estos servidores Scribe se encuentran configurados para recibir y enviara mensajes a un servidor central. Si el servidor Scribe central no está disponible el servidor Scribe local escribe los mensajes a un archivo en el disco local y los envía cuando el servidor central se encuentre disponible otra vez. El mismo, fue originalmente desarrollado por Facebook y no ha sido actualizado recientemente.

**Chukwa:**

Es un proyecto de origen libre bajo la fundación Apache Software utilizado para la obtención de datos. Asimismo, también incluye un conjunto de herramientas flexibles y potentes para la visualización, monitoreo y análisis de resultados para hacer el mejor uso de los datos obtenidos.

**Sqoop:**

Apache Sqoop es una herramienta que permite la transferencia de datos entre Hadoop y un almacenamiento estructurado, tales como las bases de datos relacionadas.

**Kafka:**

Apache Kafka es un sistema de almacenamiento publicador/subscriptor distribuido, particionado y replicado. Además, presenta características como lo son la rapidez en las lecturas y escrituras; lo anterior lo convierte en una herramienta excelente para comunicar flujos de información que se generan a gran velocidad y que deben ser gestionados por una o varias aplicaciones.

## 5.2 Frameworks de cálculo

### MapReduce

Es un método para distribuir tareas a través de múltiples nodos en el clúster. Cada nodo procesa los datos almacenados en ese nodo de manera individual. Asimismo, MapReduce, como bien lo dice su nombre, consiste de dos fases: Map y Reduce que permite automatizar la paralelización y distribución, así como una tolerancia a fallos muy baja, herramientas para monitorizar y obtener el estado. Además, es una clara abstracción para los programadores, y está escrito en Java.

En el framework de MapReduce, la ejecución de un Job, o trabajo, es controlado por dos tipos de procesos:

- Un proceso único llamado JobTracker, que coordina todos los trabajos que se ejecutan en el clúster y asigna las tareas “map y reduce” a ejecutar en los TaskTrackers.
- Una serie de procesos subordinados llamados TaskTrackers, que ejecutan las tareas asignadas e informan periódicamente el progreso a el JobTracker.

A pesar de ser un método muy potente para la distribución y ejecución de tareas, MapReduce deja ver una de sus limitaciones relacionada con la escalabilidad y causada por la tenencia de un solo JobTracker para la coordinación y asignación de tareas. Lo anterior, ya que provoca un problema de embotellamiento en un clúster. Según la compañía Yahoo, el problema surge al alcanzar un clúster de 5000 nodos y en el cual se ejecuten 40000 tareas simultáneamente. Debido a esta limitante, se deben crear clúster más pequeños lo que da como resultado que estos sean menos poderosos.

**Yarn(Yet Another Resource Negotiator)**

Es la siguiente generación de MapReduce, disponible en la versión 2.0. Nació con el fin de dividir las dos funciones que realiza el JobTracker(NameNode) en la versión de MapReduce v1. Lo cual significa tener servicios o demonios totalmente separados e independientes de forma que la gestión de recursos estaría por un lado y, por otro, la planificación y monitorización de las tareas o ejecuciones.

De esta división surgen dos procesos:

- ResourceManager (RM): Este proceso es global y se encarga de toda la gestión de los recursos.

- ApplicationMaster (AM): Este proceso se ejecuta por aplicación y se encarga de la planificación y monitorización de las tareas.

Por lo tanto, el ResourceManager y el NodeManager (NM) esclavo de cada nodo forman el entorno de trabajo, encargándose el ResourceManager de asignar y controlar los recursos entre todas las aplicaciones del sistema. Asimismo, el ApplicationMaster se encarga de la negociación de recursos con el ResourceManager y los NodeManager para poder ejecutar y controlar las tareas. Además, le solicita la obtención de recursos para poder trabajar. Esta división ayuda a que el Framework de Hadoop pueda tener un entorno de administración de recursos y aplicaciones distribuidas lo que permite implementar múltiples aplicaciones de procesamiento de datos personalizados y específicos para realizar una tarea en cuestión.

#### **Weave**

Programación simplificada de Yarn; Weave le permite a los desarrolladores explotar o utilizar el poder que brinda Yarn en Hadoop. Lo anterior, mediante el uso de un modelo simple de programación y componentes reusables para la construcción de aplicaciones y Frameworks distribuidos. Además, el modelo de programación se asemeja mucho a Java.

#### **Cloudera SDK**

Programación simplificada de MapReduce. Antiguamente conocido como Cloudera Development Kit. Es un conjunto de bibliotecas, herramientas y documentación que hacen más fácil para los desarrolladores crear sistemas en Apache Hadoop.

### **5.3 Consulta de datos en HDFS**

#### **Java MapReduce**

MapReduce nativo en Java, un trabajo (Job) MapReduce generalmente divide la entrada de datos en fragmentos independientes que son procesados por las funciones de Map ejecutadas de manera paralela en cada uno de los nodos. El Framework se encarga de ordenar los resultados de las funciones Map, que son luego ingresadas en las funciones de reduce. Por lo general, tanto la entrada como la salida de los trabajos son almacenados en sistemas de archivos (HDFS). Además, se encarga de las tareas programadas como del monitoreo de ellas y la re-ejecución de las tareas fallidas.

#### **Hadoop Streaming**

Es una herramienta que viene incorporada en la distribución de Hadoop. Esta utilidad permite crear y ejecutar trabajos de Map/Reduce con cualquier ejecutable o script como el Mapper o el Reducer. Los anteriores son ejecutables que leen la entrada de datos línea por línea (stdin) y emiten una salida de datos (stdout). Además, se encarga de emitir el trabajo al clúster apropiado y de monitorizar el progreso del trabajo hasta que el mismo se complete.

#### **Pig**

Permite escribir operaciones de MapReduce complejas usando un simple lenguaje de scripting. Pig latin (el lenguaje) define un conjunto de comandos tales como aggregate, join y sort los cuales simplifican la consulta de datos.

Además, Apache Pig se encarga de traducir el código de pig en código MapReduce para que pueda ser ejecutado dentro de Hadoop. Además, Apache Pig puede ser extendido mediante el uso de UDFs por sus siglas en inglés (funciones definidas por el usuario), las cuales pueden ser escritas en Java o algún otro lenguaje y luego ser llamadas directamente desde Pig.

### **Hive**

Le permite a los desarrolladores escribir consultas en Hive Query Language (HQL) por sus siglas en inglés, el cual es muy similar al lenguaje SQL estándar. Los datos pueden ser consultados utilizando SQL en lugar de utilizar código escrito en Java MapReduce. Asimismo, Hive puede ser ejecutado desde una interfaz de línea de comandos (conocida como Hive Shell), a partir de una base de datos de Conectividad Java (JDBC), una base de datos de conectividad libre (ODBC). Lo anterior, aprovechando los drivers Hive ODBC/ JDBC, o de lo que se llama un cliente Hive Thrift, el cual se instala en la máquina del cliente (puede ser utilizado con aplicaciones escritas en C++, Java, PHP, Python, o Ruby).

### **Stinger / Tez**

Siguiente generación de Hive; nació como una iniciativa para mejorar Hive y tener un mejor rendimiento para la realización de casos de uso (rangos de 5 a 30 segundos); como por ejemplo la exploración de Big Data, visualización, reportes parametrizados sin la necesidad de recurrir a otras herramientas. El lenguaje HiveQL siguió siendo el mismo antes y después de esta iniciativa, solo se mejoró en algunos aspectos tales como la inclusión de nuevas características analíticas como cláusula “OVER”, soporte para la creación de subqueries en la cláusula “Where” y además de asemejar más el sistema de Hive al modelo estándar de SQL. También, los cambios hechos en iniciativa han mejorado hasta en un 90 por ciento el tiempo de ejecución de las sentencias incrementando el número de tareas que Hive puede procesar en un segundo. De igual forma, la comunidad de Hive introdujo un nuevo formato de archivo (ORCFile) para proveer una forma más eficiente, moderna y de mejor rendimiento para almacenar datos de Hive. Por último, se introdujo un nuevo Runtime Framework llamado Tez el cual es utilizado para la eliminación de la latencia y limitaciones de rendimiento que tiene Hive resultado de su dependencia a MapReduce. Tez, por su parte, optimiza la ejecución de los trabajos de Hive mediante la eliminación de las tareas innecesarias, las barreras de sincronización y las lecturas y escrituras en HDFS.

### **Cloudera Search**

Buscador de texto que funciona casi en tiempo real; Cloudera Search le permite a personas no técnicas buscar y explorar información almacenada en HDFS o HBASE. Lo anterior, porque este buscador de texto no necesita que las personas tengan conocimientos en SQL estándar o habilidades en programación; ya que provee una interfaz muy completa y simple para la realización de las búsquedas.

### **Impala**

Impala fue desarrollada por Cloudera y provee la capacidad de realizar consultas en tiempo real. Impala soporta consultas de información almacenada en Hadoop HDFS y HBase (Base de datos NoSQL), de acuerdo con Cloudera, Im-



pala se encuentra entre 3 y 30 veces más rápido que Hive. Asimismo, el core de Impala trabaja bajo la licencia de Apache y es casi un estándar de SQL a través de Hive SQL. Eso significa que se queda un tanto corto con el completo soporte del ANSI SQL. Pero, Impala sí incluye controladores ODBC/JDBC y es soportado por sistemas de inteligencia de negocios. Opcionalmente, hay una suscripción propietaria, la cual agrega un módulo de administración que permite la distribución, monitoreo de Impala, así como del análisis del rendimiento de las sentencias que se realizan.

## 5.4 SQL en Hadoop / HBase

### Hive

Provee una capa de SQL sobre la distribución de Hadoop. Permite realizar sentencias de SQL en lugar de crear código Java para MapReduce.

#### Stinger / Tez

Siguiente generación de Hive.

#### Hadapt

Producto comercial que ofrece soporte para Hadoop mediante el análisis interactivo de grandes cantidades de datos y basado en SQL. Hadapt combina la arquitectura robusta y escalable de Hadoop con una capa híbrida de almacenamiento que incorpora una base de datos relacional.

#### Greenplum HAWQ

Base de datos relacional con soporte SQL que trabaja sobre Hadoop HDFS. HAWQ es un motor de consultas SQL en paralelo que combina las ventajas tecnologías de la base de datos Pivotal Analytic (Pivotal Analytic Databases) con la escalabilidad y conveniencia de Hadoop. Además, HAWQ escribe y lee datos en Hadoop HDFS de manera nativa y ofrece una interfase completa y compatible con SQL que le permite al usuario interactuar con grandes volúmenes de datos de forma muy confiable.

#### Impala

Consultas en tiempo real sobre Hadoop. Desarrollado por Cloudera.

#### Presto

Es un motor de consultas distribuidas de SQL de código abierto que sirve para la ejecución interactiva de consultas analíticas desde fuentes de datos de todos los tamaños que van desde gigabytes a petabytes. Desarrollado por Facebook.

#### Phoenix

Es un motor de consultas SQL para Apache HBase (Base de datos NoSQL). Accede a ella mediante el uso de un controlador JDBC y permite consultar y gestionar tablas HBase utilizando SQL.

#### Spire

Motor de consultas SQL para Apache HBase (Base de datos NoSQL) desarrollado por DrawnToScale.com. Fue uno de los primeros productos en combinar la escalabilidad de Hadoop con la robustez de SQL. Actualmente, este motor de consultas incorpora soporte para MongoDB.

#### Citus Data

Base de datos relacional que posee soporte para consultas SQL sobre Hadoop HDFS. Su motor de búsqueda de gran alcance paraleliza consultas SQL sobre grandes bases de datos y permite respuestas en tiempo real. Producto comercial desarrollado por CitusData.

#### **Apache Drill**

Análisis interactivo de grandes cantidades de datos. Este proyecto de Apache fue inspirado por Dremel de Google y permite a los usuarios consultar terabytes de datos en cuestión de segundos. Además, es compatible con una amplia gama de formatos de datos como Protocolos Buffers, Avro y JSON, y aprovecha Hadoop y HBase como sus fuentes de datos. DrQL es su lenguaje de consulta y el cual permite compatibilidad con con Google BigQuery.

### **5.5 Consultas en Tiempo Real**

#### **Apache Drill**

Análisis interactivo de grandes cantidades de datos.

#### **Impala**

Impala fue desarrollada por Cloudera y provee la capacidad de realizar consultas en tiempo real.

### **5.6 Stream processing (Procesamiento de flujo de datos)**

#### **Storm**

Procesamiento de flujo de datos en tiempo real.

#### **Apache S4**

Sistema de procesamiento de flujo de datos escalable y contra fallos.

#### **Samza**

Sistema procesamiento de flujo de datos open source. Samza trabaja con Apache Kafka y Apache Yarn para proporcionar un framework de procesamiento de flujo en tiempo real.

### **5.7 Almacenamiento NoSQL**

#### **HBase**

Orientadas a columnas. Escrita en Java y mantenida por el proyecto Hadoop de Apache. Además, se utiliza para procesar grandes cantidades de datos.

#### **Cassandra**

Orientadas a columnas; Utiliza un modelo híbrido, entre orientada a columnas clave-valor.

#### **Redis**

Almacenamiento clave-valor. Desarrollada en C y de código abierto, es utilizada por y Stack Overflow (a modo de caché).

#### **DynamoDB**

Almacenamiento clave-valor. Desarrollada por Amazon, es una opción de almacenaje se puede usar desde los Amazon Web Services. La emplean el Washington Post y Scopely.

**MongoDB**

Orientada a documentos. Probablemente la base de datos NoSQL, en la actualidad, es la mas famosa.

**CouchDB**

Orienta a documentos de Apache. Una de sus interesantes características es que los datos son accesibles por medio de una API Rest.

**Infinite Graph**

Orientada a grafos. Escrita en Java y C++ por la compañía Objectivity. Tiene dos modelos de licenciamiento: uno gratuito y otro de pago.

**Neo4j**

Orientada a grafos. Base de datos de código abierto, escrita en Java por la compañía Neo Technology.

**5.8 Hadoop en la Nube****Amazon Elastic MapReduce (EMR)**

Framework de Hadoop en la nube para la administración y procesamiento de grandes cantidades de datos de una manera fácil por medio de instancias de Amazon EC2 y de una escalabilidad dinámica. Además, Amazon EMR permite la ejecución de otros frameworks populares como Spark y Presto así como la interacción de datos con otros almacenamientos de datos en AWS como los son Amazon S3 y Amazon DynamoDB.

**Hadoop on Rackspace**

Nube privada desarrollada por OpenStack. Compuesto por un software libre y de código abierto llamado Alamo, el cual permite ejecutar software de control en la nube como por ejemplo OpenStack para el manejo de almacenamiento, clústers y recursos de red en un centro de datos y utilizando a través de un dashboard o vía OpenStack API.

**Hadoop on Google Cloud**

Plataforma de Google en la nube que permite el procesamiento de datos de forma fácil y utilizando Hadoop. Esta plataforma en la nube esta compuesta por un conjunto de bibliotecas de software y la infraestructura optimizada de Google. Lo anterior, permite procesar Big Data con un alto rendimiento. Además, la ejecución de trabajos de MapReduce pueden ser ejecutados directamente en los datos utilizando Google Cloud Storage y sin la necesidad de copiarlos en el disco local median HDFS. Por su parte, Hadoop en Cloud Platform Google también proporciona conectores que le permiten acceder a los datos almacenados en BigQuery y almacén de datos, así como Google Cloud Storage.

**Whirr**

Forma fácil de manejar clústers de Hadoop en servicios en la nube como Amazon y RackSpace. Apache Whirr esta compuesto por un conjunto de librerías que permiten la ejecución de servicios en la nube de una forma neutra sin preocuparse por los proveedores. Además, permite la ejecución de sistemas rápidamente mediante el uso de configuración predefinidas y a la misma vez con la opción de cambiar esas configuraciones una vez requerido

## 5.9 Herramientas de Flujo de Trabajo /Panificadores (Work flow Tools / Schedulers)

### Oozie

Integrado bajo los proyectos que conforman Hadoop. Una vez funcionando un sistema para el procesamiento de Big Data mediante el uso de Hadoop, Oozie se encarga de simplificar los flujos de trabajo y coordinar estos procesos para un correcto análisis; ya que los mismos se ejecutan en distintos momentos.

#### Cascading

Es un framework para desarrolladores, analistas de datos, permite desarrollar análisis de datos robustos y aplicaciones de gestión de datos sobre Hadoop.

#### Scalding

Es utilizado por Twitter para el análisis de datos y machine learning, particularmente en los casos en los que se necesita más que consultas SQL en los logs, para el ajuste de modelos de instancia y el procesamiento de matriz.

## 5.10 Serialization Frameworks

### Avro

Es servicio de serialización que forma parte de los proyectos de Hadoop. Utiliza JSON para definir tipos de datos y protocolos; además de serializar datos en formato binario. Seguidamente, el esquema JSON que define el archivo es guardado dentro del mismo archivo; lo anterior lo da como resultado una mayor facilidad a las aplicaciones a la hora de leer dicho archivo.

#### Trevni

Está integrado en Impala. Trevni es un nuevo formato de almacenamiento de datos en columnas que brinda un rendimiento superior para la lectura de grandes conjuntos de datos almacenados en columnas.

## 5.11 Sistemas de Monitoreo

### Hue

Desarrollado por Cloudera, es una interfaz de usuario web libre que permite gestionar Apache Hadoop y su ecosistema de forma fácil y amigable.

#### Ganglia

Sistema de monitoreo libre utilizado en sistemas de alto rendimiento como clústers y Grids. Ganglia proporciona agentes que recogen indicadores de salud en el clúster como la utilización del CPU, uso de disco duro, ejecutando tareas en máquinas individuales y luego enviando estadísticas a un punto central que se encarga de mostrar a los administradores la salud global del clúster.

#### Open TSDB

Es una aplicación que utiliza HBase como su base de datos para almacenar y recuperar información de una forma rápida. OpenTSDB, por su parte permite generar gráficos, promedios, etc. Lo anterior, mediante el uso de una interfaz web la cual permite tener como origen de datos cualquier lugar como un servidor o aplicación que permita crear una conexión TCP a OpenTSDB.

**Nagios**

Es una aplicación libre de monitoreo para redes de computación la cual permite notificar sobre errores que ocurran y necesiten ser arreglados de forma rápida. Nagios fue principalmente diseñado para ser usado en sistemas operativos Linux, pero, actualmente funciona bajo sistemas Unix y sistemas basados en Unix.

**5.12 Plataformas / Aplicaciones****Mahout**

Es un software de aprendizaje de máquinas que le permite a aplicaciones analizar grandes cantidades de datos. Además, está compuesto por tres técnicas de aprendizaje:

- Recomendación: Este algoritmo permite analizar nuestra información en conjunto con la de los demás para predecir cuáles nuevos ítemes nos gustarían o no según las preferencias que hayamos tenido en ítemes pasados.
- Clasificación: Este algoritmo permite clasificar información según la categoría, por ejemplo un correo electrónico marcado como spam, influirá en el motor de clasificación del correo de forma que los correos nuevos ingresados ya podrán ser clasificados a partir de la información que contenga el motor evitando se produzca spam en el futuro.
- Clustering: Forma grupos de datos similares basado en características comunes. Por ejemplo, Google News utiliza este algoritmo para hacerle frente a los artículos siempre cambiantes alrededor del mundo y poder tener al día los artículos más recientes.

**Giraph**

Es un procesador de grafos utilizado en Hadoop v2 para realizar tareas que no se ajustan a MapReduce y con un mejor rendimiento. Giraph en lugar de implementar funciones de Map y Reduce, utiliza un vértice el cual posee un valor y un borde capaz de enviar y recibir mensajes a otros vértices en el grafo mientras se realizan las interacciones de cálculos. Asimismo, es usado por empresas como Facebook y PayPal, para ayudar a representar y analizar miles de millones (o incluso billones) de conexiones a través de grandes conjuntos de datos. También, fue inspirado por el framework Pregel de Google y puede ser integrado de forma fácil con Apache Accumulo, Apache HBase, Apache Hive, y Cloudera Impala.

**Lily**

Unifica Apache HBase, Hadoop y Solr en una plataforma interactiva de datos ampliamente integrada con las API de fácil uso de acceso, un lenguaje de modelo de datos de alto nivel y el esquema, flexibles, la indexación en tiempo real y la potencia de búsqueda expresiva de Apache Solr. Lo mejor de todo, Lily es de código abierto, permitiendo que cualquiera pueda explorar y aprender lo que Lily puede hacer.

### 5.13 Distributed Coordination

#### Zookeeper

Es un servicio de coordinación de alto rendimiento para aplicaciones distribuidas. Además, ofrece una serie de servicios como la gestión de configuraciones, naming, sincronización, grupos de servicios sin la necesidad de crearlos desde cero; ya que ofrece el uso de una interfaz simple.

#### Book keeper

Subproyecto de Zookeeper compuesto por un servicio distribuido de logs llamado BookKeeper y un sistemas publicador/subscriptor distribuido construido sobre BookKeeper llamado Hedwig.

### 5.14 Herramientas de Análisis de Datos

#### R language

R es un entorno de software libre para computación y gráficos estadísticos. Compila y ejecuta en una amplia variedad de plataformas UNIX, Windows y MacOS.

#### RHIPE

Integra "R" y Hadoop. En un análisis de "D y R", los datos se dividen en subconjuntos, formando múltiples divisiones. Luego los métodos numéricos y de visualización se aplican a cada uno de los subconjuntos de una división, y los resultados de cada método se recombinan a través de subconjuntos.

### 5.15 Procesamiento Distribuido de Mensajes (Distributed Message Processing)

#### Kafka:

Apache Kafka es un sistema de almacenamiento publicador/subscriptor distribuido, particionado y replicado.

#### Akka:

Es un conjunto de herramientas y runtimes para la construcción altamente concurrente, distribuida, y resistente de aplicaciones controladas por mensajes en la JVM.

#### RabbitMQ

Es un sistema robusto de mensajería para aplicaciones fácil de usar y con la capacidad de ser ejecutado en los principales sistemas operativos. Además, RabbitMQ soporta una gran cantidad de plataformas de desarrollo y está licenciado bajo licencias libres y propietarias.

### 5.16 Herramientas de Inteligencia de Negocios

#### Datameer

Primera plataforma de análisis de Big Data para Hadoop-as-a-Service. Fue desarrollado para simplificar el análisis de Big Data utilizando una sola aplicación, sobre la plataforma Hadoop. Datameer también fue desarrollado para

funcionar de forma nativa en todas las distribuciones de Hadoop y aprovechar la escalabilidad y poder de cálculo del clúster de Hadoop.

#### **Tableau**

Es la forma mas rápida y fácil de compartir análisis en las nube. Se encuentra alojada en el servidor Tableau y funciona como la versión SaaS (Software como servicio). Esta herramienta de inteligencia de negocios permite publicar dashboards mediante el uso de Tableau Desktop y a la misma vez permite compartirlo con clientes o colegas.

#### **Pentaho**

Es una plataforma de inteligencia de negocios ampliamente utilizada como herramientas BI de código libre. Esta proporciona características como reporting intuitivo, análisis OLAP, cuadros de mando, integración de datos, minería de datos. El modelo de negocio de código libre y comercial de Pentaho soporte, servicios y mejoras del producto vía suscripciones anuales

#### **SiSense**

Esta herramienta de BI permite crear Dashboards interactivos utilizados para el análisis de Big Data. El análisis de grandes cantidades de datos se realiza mediante un interfaz *drag y drop* que permite construir los Dashboards interactivos fácilmente.

### **5.17 Librerías / Frameworks**

#### **Elephant Bird**

Es un Framework utilizado para trabajar con datos estructurados en el ecosistema de Hadoop. Principalmente para cargar diferentes formatos de archivos como por ejemplo un archivo JSON a pig.

#### **Apache Crunch**

Este proyecto de apache ayuda a descomponer los problemas de procesamientos complejos en conceptos simples que pueden ser utilizados en frameworks, tales como Hadoop y Spark. Apache Crunch está siendo utilizado como una parte integral en la construcción de pipelines de procesamiento de datos para la salud que permitan un rápido desarrollo de nuevas soluciones y arquitecturas.

#### **Apache DataFu**

Es una colección de funciones definidas por el usuario útiles para el análisis de datos en Apache Pig.

### **5.18 Manejo de Datos (Data Management)**

#### **Apache Falcon**

Es un framework para la gestión del ciclo de vida de los datos para Hadoop. Simplifica la gestión de datos, además maneja los flujos de trabajo, la replicación y provee la abstracción de datos.

## 5.19 Seguridad

### Apache Sentry

Es un mecanismo de autorización unificada el cual permite almacenar datos sensibles en Hadoop. Sentry es un componente totalmente integrado que proporciona autorización y el control de acceso basado en roles a lo largo de un único sistema.

### Apache Knox

Es un API REST para interactuar con el clúster de Hadoop y el cual permite un único punto de acceso para todas las interacciones REST en el clúster. Knox es capaz de proporcionar funcionalidades valiosas para ayudar en el control, la integración, el seguimiento y la automatización de las necesidades administrativas y análisis críticos de la empresa, como por ejemplo: autenticación LDAP y proveedor de autenticación para los directorios de activos y su uso en auditorías.

## 5.20 Frameworks de Pruebas

### MrUnit

Frameworks de pruebas utilizado para Java MapReduce.

### PigUnit

Framework de pruebas utilizado para ejecutar scripts de Apache Pig. Puede ser ejecutado de forma local por lo tanto, sin la necesidad de configurar un clúster.

## 6 Casos de uso

La aparición del término Big Data en la actualidad, permite analizar la creación de soluciones eficientes para el procesamiento y almacenamiento de grandes volúmenes de información. Estas soluciones o arquitecturas dependen del caso de uso que se vaya a desarrollar y tienen como objetivo la reducción de tiempos y costos los cuales brinden mejoras a los procesos internos de las compañías.

### 6.1 Caso de uso uno

#### Recomendaciones de productos

En nuestro primer caso “(JAGADISH et al., 2014)” cubriremos una práctica muy común la cual llamaremos recomendaciones de productos. Muchos comercios en línea, actualmente tratan de conocer las preferencias de sus clientes colectando toda la información posible de los mismos, por ejemplo en este caso mediante las búsquedas que algún cliente haya realizado, el comercio es capaz de coleccionar tanto los resultados de las búsquedas como los artículos en los que el cliente mostró más interés por medio de clics. Una vez coleccionada esta información se procesa para poder obtener algunos artículos relacionados en los cuales el cliente podría estar interesado, estas luego se le muestran al cliente mediante los sitios web que este frecuente o social media.





Fig. 8. Caso de Uso 1.

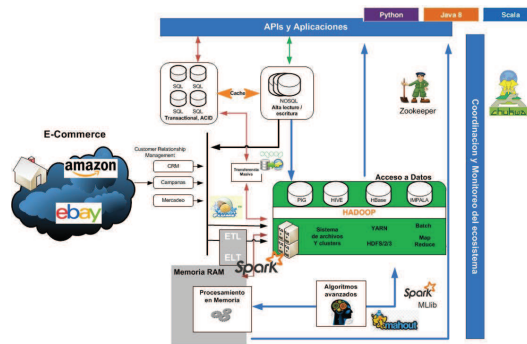


Fig. 9. Arquitectura caso de uso 1

**Flujo de datos**

Los datos entran al *pipeline* directo a Hadoop o bien pueden pasar por Spark como ETL para ser procesados en memoria junto con datos que ya estén en Hadoop, estos pueden ser extraídos de disco por medio de Spark y mantenerlos en memoria para mayor velocidad y eficiencia al procesarlos. Una vez procesados los datos se pueden desplegar al usuario mediante las APIs desarrolladas en Python, Java o Scala y almacenarlos en Hadoop. Además, se contempla la necesidad de extraer datos de bases de datos Transaccionales; como por ejemplo información de inventario, productos, ID de clientes etc. Esto gracias a la aplicación SQOOP. Asimismo, se contempla también utilizar herramientas como PIG, HIVE, HBase, IMPALA para acceder y procesar los datos.

**6.2 Caso de uso dos**

**Selección de rutas**

En el siguiente caso de uso “(JAGADISH et al., 2014)” se pretende tomar datos de GPS; ya sea de automóviles o de usuarios con su teléfono, también se pueden tomar en cuenta datos de sensores en las calles, así como información proveniente de semáforos inteligentes. Esta se debe procesar en tiempo real con el fin de obtener un cálculo estimado de por cuál camino sería más conveniente avanzar para llegar más rápido a algún lugar.



Fig. 10. Caso de Uso 2.

### Flujo de datos

En nuestra arquitectura los datos entran por Spark Streaming y son procesados en RDD con frecuencia de 1 segundo por batch, una vez procesados llegan a Spark para unirse con información previamente llamada de Hadoop para así permitir correr cálculos en RAM y enviarlos a las APIs que presentarán los resultados.

## 7 Conclusiones

Una vez concluido el planteamiento de una arquitectura de Big Data Analytics a partir del estudio de las propuestas de la industria y la academia, se concluyó lo siguiente:

A nuestro parecer las tecnologías *open source* han logrado un avance positivo en el campo de Big Data. El aporte de las diferentes compañías ha sido importante para lograr superar los obstáculos y retos que se han encontrado en el camino. Además, las grandes compañías no solo han sacado provecho de esto; sino también han hecho valiosos aportes evolucionando así tanto el core de Hadoop como las diferentes herramientas que forman el ecosistema, como es el caso de Yarn, Spark streaming, HDFS2 y HDFS3, etc.

Actualmente, se habla que los datos son el nuevo recurso natural. Esta relación se da porque las empresas cada vez más dependen del conocimiento que

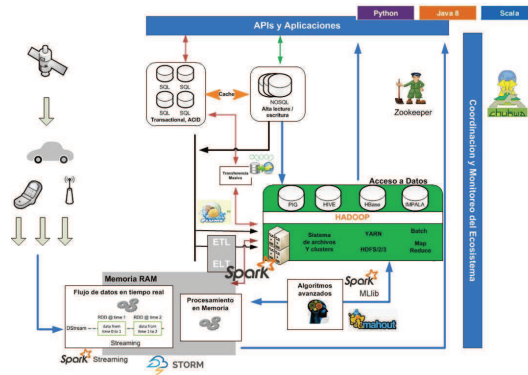


Fig. 11. Arquitectura caso de uso 2..

los datos generan para poder competir en el mercado. Asimismo, este conocimiento es de vital importancia para las empresas; ya que les permite producir innovación y crear nuevas líneas de negocio.

Otro aspecto a tener en cuenta, es que la tendencia no solo se basa en procesar la gigantesca cantidad de datos que generamos, sino también, se requiere interactuar con ella en tiempo real. Por lo tanto, las tecnologías de streamings están teniendo tanto auge en los sistemas actuales.

De esta manera, concluimos nuestra investigación; pero, no sin antes recalcar la importancia de usar un sistema para el tratamiento de Big Data en los diferentes sectores de la industria que permita trabajar con grandes volúmenes de datos, de la manera más rápida y eficiente posible, también de poder adaptarse a todos los formatos estructurados o no existentes y futuros.

## References

- Apache, F. (n.d.). *Hadoop*. pages 12
- Chunmei Duan1, q. (2014). Design of big data processing system architecture based on hadoop under the cloud computing. *Applied Mechanics and Materials*(556-562), 6302 - 6306. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=96393594&lang=es&site=ehost-live> pages 4, 6
- Cloudera, I. (n.d.). *Hadoop for the enterprise*. pages 9
- Editor, E. (n.d.). *Experfy insights*. pages 10
- Gorkahurtado. (n.d.). *Big data y hadoop. cloudera vs hortonworks*. pages 8, 9
- Hortonworks. (n.d.). *Do hadoop. everywhere*. pages 13
- IBM. (n.d.). *Infosphere biginsights v3.0*. pages 7
- JAGADISH, j., H.V.1, GEHRKE, j., JOHANNES2, LABRINIDIS, . 1., ALEXANDROS3, PAKONSTANTINOU, y., YANNIS5, PATEL, J. M., RAMAKRISHNAN, r., RAGHU7, & SHAHABI, . s., CYRUS8. (2014). Big data and its technical challenges. *Communications of the ACM*, 57(7),

86 - 94. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=96868411&lang=es&site=ehost-live> pages 24, 26

Oracle. (n.d.). *Oracle big data*. pages 10