



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA COMPUTACIÓN

LAS REDES SOCIALES Y EL
ALMACENAMIENTO DE SU ESTRUCTURA
SOBRE PLATAFORMAS DE GRANDES
VOLÚMENES DE DATOS PARA LA
REALIZACIÓN DE ANÁLISIS

TRABAJO DE SEMINARIO PRESENTADO ANTE LA ILUSTRE

UNIVERSIDAD CENTRAL DE VENEZUELA POR EL

BR. ERIC GABRIEL BELLET LOCKER.

C.I. 24.463.483.

TUTORES: JESÚS LARES Y JOSÉ SOSA.

CARACAS, ABRIL 2016.

Resumen

En el siguiente trabajo escrito se describen diversos conceptos generales muy utilizados en el área de la computación, los cuales son importantes de manejar. El principal tema a tratar corresponde a Ciencias de los Datos, específicamente Big Data, el cual se relaciona con diferentes tecnologías, como Apache Hadoop. Se señalan con una breve explicación, algunas herramientas y distribuciones de este framework.

Asociado al tema Big Data, se define uno de los campos en el cual es utilizado, las Redes Sociales, su descripción, clasificación y las ventajas de analizarlas. Es evidente la gran cantidad de datos que manejan las redes sociales, y gracias a Big Data es posible realizar análisis sobre esta, en consecuencia en este documento se establece posibles módulos, arquitectura e indicadores relacionados.

Palabras clave: Big Data, Redes Sociales, Apache Hadoop.

Índice general

Resumen	2
Introducción	9
1. MARCO TEÓRICO	10
1.1. Dato	10
1.2. Información	10
1.3. Conocimiento	10
1.4. Ciencias de datos	11
1.5. Grandes volúmenes de información (Big Data)	11
1.5.1. Definición de grandes volúmenes de información	11
1.5.2. Campos en los cuales es utilizado	13
1.6. Base de datos	14
1.6.1. Base de datos relacionales	15
1.6.1.1. Ventajas de las base de datos relacionales	15
1.6.1.2. Desventajas de las base de datos relacionales	15
1.6.2. Base de datos no relacionales	15
1.6.2.1. Ventajas de las base de datos no relacionales	16
1.6.2.2. Desventajas de las base de datos no relacionales	16
1.6.2.3. Tipos de base de datos no relacionales	17
1.6.2.3.1. Orientada a columnas	17
1.6.2.3.2. Orientada a clave/valor	18
1.6.2.3.3. Orientada a documentos	18
1.6.2.3.4. Orientada a grafo	18
1.7. Almacén de datos (Data Warehouse)	19
1.8. Inteligencia de negocios	19
1.9. Lenguajes de programación	19
1.9.1. R	20
1.9.1.1. R STUDIO	20
1.9.2. JAVA	22
1.9.3. PYTHON	23
1.9.4. JAVASCRIPT	24
1.9.4.1. NODE.JS	24
1.9.4.2. ANGULAR.JS	24

1.9.4.3.	D3.JS	24
1.10.	APACHE HADOOP	25
1.10.1.	Definición de APACHE HADOOP	25
1.10.1.1.	COMMONS	25
1.10.1.2.	MAPREDUCE	25
1.10.1.2.1.	JobTracker	26
1.10.1.2.2.	TaskTracker	26
1.10.1.2.3.	Map	26
1.10.1.2.4.	Reduce	27
1.10.1.3.	HADOOP DISTRIBUTED FILE SYSTEM (HDFS)	27
1.10.1.3.1.	Características de HDFS	28
1.10.1.3.2.	Arquitectura de HDFS	28
1.10.1.4.	Yet Another Resource Negotiator (YARN)	29
1.10.1.4.1.	Arquitectura de YARN	30
1.10.2.	Herramientas del ecosistema	30
1.10.2.1.	Administración de los datos	31
1.10.2.2.	Acceso a los datos	31
1.10.2.2.1.	APACHE SPARK	31
1.10.2.2.2.	APACHE ACCUMULO	36
1.10.2.2.3.	APACHE HBASE	37
1.10.2.2.4.	APACHE HIVE	39
1.10.2.2.5.	APACHE STORM	40
1.10.2.2.6.	APACHE MAHOUT	41
1.10.2.3.	Integración	41
1.10.2.3.1.	APACHE FALCON	41
1.10.2.3.2.	APACHE FLUME	42
1.10.2.3.3.	APACHE SQOOP	42
1.10.2.4.	Seguridad	43
1.10.2.4.1.	APACHE KNOX	43
1.10.2.4.2.	APACHE RANGER	43
1.10.2.5.	Operaciones	43
1.10.2.5.1.	APACHE AMBARI	44
1.10.2.5.2.	APACHE ZOOKEEPER	44
1.10.2.5.3.	APACHE OOZIE	44
1.10.3.	Distribuciones HADOOP	45
1.10.3.1.	CLOUDERA	45
1.10.3.2.	MAPR	46
1.10.3.3.	HORTONWORKS	46
2.	REDES SOCIALES	48
2.1.	Concepto de Red Social	48
2.2.	Clasificación de las redes sociales	48
2.2.1.	Por su público objetivo y temática	48
2.2.2.	Por el sujeto principal de la relación	49
2.2.3.	Por su localización geográfica	49
2.2.4.	Por su plataforma	50

2.3.	Análisis de Redes Sociales	50
2.4.	Arquitectura Big Data para Redes Sociales	51
2.4.1.	Captación	51
2.4.2.	Almacenado	51
2.4.3.	Análisis	52
2.4.4.	Visualización	52
2.5.	Módulos del sistema Big Data para Redes Sociales	53
2.5.1.	Módulo I: Análisis de Redes Sociales	53
2.5.1.1.	Objetivo principal del Análisis de Redes Sociales	53
2.5.1.2.	Principales Funciones para el Análisis de Redes Sociales	53
2.5.1.2.1.	Recopilación de datos	53
2.5.1.2.2.	Identificación de comunidades	53
2.5.1.2.3.	Identificación de usuarios influyentes	54
2.5.1.2.4.	Análisis de tendencias	56
2.5.1.3.	Métricas	56
2.5.1.3.1.	Principales métricas para el Análisis de Redes Sociales	57
2.5.2.	Módulo II: Monitoreo de Redes Sociales	57
2.5.2.1.	Objetivo Principal del Monitoreo de Redes Sociales	57
2.5.2.2.	Principales Funciones para el Monitoreo de Redes Sociales	58
2.5.2.2.1.	Monitoreo Inteligente	58
2.5.2.2.2.	Medición de usuarios nacionales	58
2.5.2.2.3.	Medios locales y mundiales	58
2.5.2.2.4.	Selección de temas importantes	58
2.5.2.2.5.	Alertas y reportes automáticos	58
2.6.	Indicadores Big Data para Redes Sociales	59
2.6.1.	Retorno de la inversión en Social Media (ROI)	59
2.6.2.	Compromiso (ENGAGEMENT)	59
2.6.3.	Valor de la marca (BRAND ASSET VALUATION)	59
2.6.4.	Reacciones de consumo (CONSUMER REACTIONS)	60
2.6.5.	Seguidores (BRAND AUDIENCE)	61
2.6.6.	Influyentes (INFLUENCERS)	61
2.6.7.	Contenidos	62
2.6.8.	Localización	62
2.7.	Redes sociales en Venezuela	63
3.	Análisis de redes sociales	65
3.1.	Análisis de redes sociales o social network analysis (SNA)	65
3.2.	Pasos para realizar un análisis de redes sociales	65
3.2.1.	Estudiando las características generales	65
3.2.1.1.	Redes de mundo pequeño (small world networks)	66
3.2.1.2.	Redes libres de escala (scale free networks)	66
3.2.2.	Estudiando la posición de los actores	66

3.2.2.1.	Centralidad de grado (degree centrality)	66
3.2.2.2.	Centralidad de Bonacich	66
3.2.2.3.	Centralidad de vector propio (eigenvector centrality)	67
3.2.2.4.	Centralidad de cercanía (closeness centrality) . .	67
3.2.2.5.	Centralidad de intermediación (betweenness centrality)	67
3.2.3.	Detección de comunidades	67
3.2.3.1.	Técnicas de detección	68
3.2.4.	Visualización	68
4.	Problema	69
4.1.	Planteamiento del problema	69
4.2.	Justificación del problema	70
4.2.1.	¿Por qué es un problema?	70
4.2.2.	¿Para quién es un problema?	70
4.2.3.	¿Desde cuándo es un problema?	70
4.3.	Objetivos de la investigación	70
4.3.1.	Objetivo general	70
4.3.2.	Objetivos específicos	70

Índice de figuras

1.1.	Figura 1: Proceso de ciencias de los datos.	11
1.2.	Figura 2: IDE RSTUDIO.	21
1.3.	Figura 3: Hola mundo en Java.	22
1.4.	Figura 5: Funcionamiento de MapReduce.	27
1.5.	Figura 4: Arquitectura HDFS.	29
1.6.	Figura 5: Arquitectura Yarn.	30
1.7.	Figura 7: DAG (Directed Acyclic Graph).	32
1.8.	Figura 8: Proceso Resilient Distributed Dataset.	34
1.9.	Figura 9: Narrow transformation y Wide transformation.	35
1.10.	Figura 10: Arquitectura Apache Accumulo.	37
1.11.	Figura 11: Modelo utilizando Apache HBase.	38
1.12.	Figura 12: Arquitectura Apache Hive.	40
1.13.	Figura 13: Sqoop.	42
1.14.	Figura 14 : Zookeeper.	44
1.15.	Figura 15 : Cloudera.	45
1.16.	Figura 16 : MapR.	46
1.17.	Figura 17 : Hortonworks.	47
2.1.	Figura 18 : Ejemplo de una arquitectura Big Data para redes sociales.	51
2.2.	Figura 19 : Valor de la marca.	60
2.3.	Figura 20 : Análisis por categorías.	62
2.4.	Figura 20 : Estadísticas de redes sociales en Venezuela.	63
2.5.	Figura 21 : Estadísticas de redes sociales en Venezuela 2.	64

Índice de tablas

1.1. Tipos de Base de Datos NoSql.	17
1.2. Comparación entre HBase y HDFS.	39

Introducción

A medida que pasa el tiempo algunos de los problemas en la computación cambian gracias a los avances científicos, en el pasado uno de los problemas era el almacenamiento de grandes volúmenes de datos, en el presente el problema es detectar oportunidades o valor en estos datos, en consecuencia las empresas cada vez se preocupan más de la obtención y recolección de datos. Las propiedades de los datos han evolucionado, donde el volumen de estos ha incrementado, ha incrementado su diversidad en forma y origen, es decir la variedad, ha aumentado la necesidad de obtener resultados en tiempo real, en otras palabras la velocidad, el valor es una propiedad importante para las empresas, igual que la visualización de análisis de los datos, lo que aporta criterios para la toma de decisiones. La gestión de los datos se consigue gracias a los sistemas Big Data.

Las Redes Sociales son un ejemplo claro de grandes volúmenes de datos, y realizar distintos tipos de análisis sobre estas puede ser muy útil. Muchas organizaciones se han dado cuenta que la gestión y un análisis completo de los datos, puede influir positivamente en la empresa si se realiza de forma adecuada y se toman las decisiones correctas. Big Data hace posible el análisis sobre la gran cantidad de datos que generan las redes sociales.

CAPÍTULO 1

MARCO TEÓRICO

1.1. Dato

Los datos son números, letras o símbolos que describen objetos, condiciones o situaciones. Son el conjunto básico de hechos referentes a una persona, cosa o transacción de interés para distintos objetivos, entre los cuales se encuentra la toma de decisiones. [1]

1.2. Información

La información es un sistema de control, en tanto que es la propagación de consignas que deberíamos de creer. En tal sentido la información es un conjunto organizado de datos capaz de cambiar el estado de conocimiento. Un grupo de datos ordenados y supervisados, que sirven para construir un mensaje basado en un cierto fenómeno o ente, la cual permite resolver problemas y tomar decisiones, es información, ya que su aprovechamiento racional es la base del conocimiento. [2]

1.3. Conocimiento

Fidias Arias (2004), define el conocimiento como un "proceso en el cual se relacionan el sujeto que conoce, que percibe mediante sus sentidos, y el objeto conocido y percibido". El conocimiento es el acto o efecto de conocer. Es la capacidad del hombre para comprender por medio de la razón la naturaleza, cualidades y relaciones de las cosas.

La ciencia considera que, para alcanzar el conocimiento, es necesario seguir un método. El conocimiento científico no sólo debe ser válido y consistente desde el punto de vista lógico, sino que también debe ser probado mediante el método científico o experimental. [3]

1.4. Ciencias de datos

Ciencias de los datos es un campo interdisciplinario, el cual abarca un conjunto de procesos y sistemas para extraer información de un conjunto de datos estructurados o no estructurados con el objetivo de generar conocimiento. Esta área está estrechamente relacionada con las estadísticas, la minería de datos, análisis predictivo, entre otros. La ciencia de datos utiliza la preparación de datos, estadísticas, modelos predictivos y de aprendizaje automático para investigar problemas en diversos ámbitos. [4]

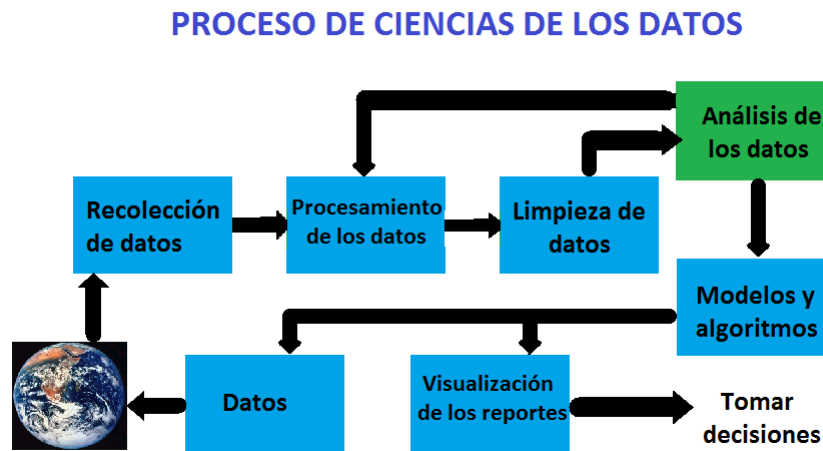


Figura 1.1: Proceso de ciencias de los datos.

1.5. Grandes volúmenes de información (Big Data)

1.5.1. Definición de grandes volúmenes de información

Los grandes volúmenes de información (Big Data) es un concepto que hace referencia a la acumulación de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos. Se define como un conjunto de herramientas informáticas destinadas a la manipulación, gestión y análisis de grandes volúmenes de datos de todo tipo, los cuales no pueden ser gestionados por las herramientas informáticas tradicionales, que tiene por objetivo analizar datos e información de manera inteligente que ayuden a una correcta toma de decisión.

En la actualidad la cantidad de información digital que se genera diariamente en nuestro planeta crece exponencialmente, lo cual ha desencadenado que muchas empresas y organizaciones, desean utilizar esta información con el objetivo de mejorar las prestaciones de sus servicios o negocios. Por lo tanto, el objetivo fundamental del big data es dotar de una infraestructura tecnológica a las empresas y organizaciones con la finalidad de poder almacenar, tratar y analizar de manera económica, rápida y flexible la gran cantidad de datos que se generan diariamente, para ello es necesario el desarrollo y la implantación tanto de hardware como de software específicos que gestionen esta explosión de datos con el objetivo de extraer valor para obtener información útil para nuestros objetivos o negocios. [5]

El término de Big Data, en general se definen con 8 dimensiones llamada las 8V [6], Velocidad, Veracidad, Valor, Volumen, Variedad, Variabilidad, Visualización y Visión:

- Velocidad: La tecnología Big Data ha de ser capaz de almacenar y trabajar en tiempo real con las fuentes generadoras de información como sensores, cámaras de vídeos, redes sociales, blogs, páginas webs, etc, fuentes que generan millones y millones de datos al segundo, por otro lado la capacidad de análisis de dichos datos han de ser rápidos reduciendo los largos tiempos de procesamiento que presentaban las herramientas tradicionales de análisis.
- Veracidad: Big Data ha de ser capaz de tratar y analizar inteligentemente este vasto volumen de datos con la finalidad de obtener una información verídica y útil que nos permita mejorar nuestra toma de decisiones.
- Volumen de datos: Como su propio nombre indica la tecnología Big Data (datos masivos) ha de ser capaz de gestionar un gran volumen de datos que se generan diariamente por las empresas y organizaciones de todo el mundo.
- Variedad de datos: Big data ha de tener la capacidad de combinar una gran variedad de información digital en los diferentes formatos en las que se puedan presentar ya sean en formato vídeo, audio o texto.
- Valor: El valor se refiere a nuestra capacidad de convertir los datos en valor. Es importante que las empresas realizar cualquier intento de recoger y aprovechar los datos grandes. Todos los que los datos disponibles crearán mucho valor para las organizaciones, sociedades y consumidores. Grandes datos significa un gran negocio y todas las industrias cosecharán

los beneficios de los grandes datos. Por supuesto, los datos en sí mismo no es valioso en absoluto. El valor está en los análisis realizados en esos datos y cómo los datos se convierte en información y, finalmente, convirtiéndola en conocimiento. El valor está en cómo las organizaciones usarán esos datos y convertir su organización en una empresa centrada en la información que se basa en ideas derivadas de los análisis de datos para la toma de decisiones.

- **Variabilidad:** La variabilidad se refiere a los datos cuyo significado está en constante cambio. Este es particularmente el caso cuando la recolección de datos se basa en el procesamiento del lenguaje.
- **Visualización:** Esta es la parte dura de grandes volúmenes de datos, hacer que la gran cantidad de datos sea comprensible de manera que sea fácil de entender y leer. Con los análisis y visualizaciones adecuadas, los datos brutos se pueden utilizar para tomas de decisiones adecuadas. Las visualizaciones por supuesto no significan gráficas ordinarias o gráficos circulares. Significan gráficos complejos que pueden incluir muchas variables de datos sin dejar de ser comprensible y legible. La visualización puede no ser la parte más difícil con respecto al uso de la tecnología, pero lograr buenos resultados seguro que es la parte más difícil. Contar una historia compleja en un gráfico es muy difícil, pero también es extremadamente crucial.
- **Visión:** todas las empresas, que se inician con grandes volúmenes de datos deben tener una visión, qué hacer con ellos. La visión define las metas que se pretenden conseguir en el futuro. Estas metas tienen que ser realistas y alcanzables, puesto que la propuesta de visión tiene un carácter inspirador y motivador. La descarga de Hadoop, la instalación de él y la alimentación con algunos datos no ayudarán. La empresa tiene que estar lista para la transformación digital. Si la organización no entiende lo que puede ofrecer grandes volúmenes de datos, no habrá ningún éxito.

1.5.2. Campos en los cuales es utilizado

El manejo de grandes volúmenes de datos es utilizado y se puede utilizar en múltiples áreas, por ejemplo:

- **Seguridad:** Su potencial reside en la capacidad de análisis de volúmenes de datos antes impensable de una manera óptima y ágil. Existen, por ejemplo, modelos de análisis del comportamiento humano para prevenir atentados terroristas obtenidos mediante un análisis permanente de las cámaras, sensores y accesos secuenciales a un sistema.
- **Investigación médica:** La investigación médica puede mejorar muchísimo si es capaz de asimilar una enorme cantidad de datos (monitoreización,

historiales, tratamientos, etc.) y estructurarlos para el establecimiento de diagnósticos o la síntesis de medicamentos.

- Gobierno y toma de decisiones: Big Data ofrece una mejora y optimización en los procesos de toma de decisiones de empresas y gobiernos, permitiendo entre muchas otras, el soporte a la toma de decisiones, siendo complementario a las plataformas de “Business Intelligence” (BI).
- Internet 2.0: genera una gran multitud de datos que difícilmente se podrían gestionar sin un Big Data. Las redes sociales cada vez se extienden a más ámbitos de nuestra sociedad
- CRM: La gestión de la relación de una empresa con sus clientes suele implicar la gestión de Almacén de Datos y la interrelación de diversidad de datos (comercial, operaciones, marketing,...), diversos canales (web, redes sociales, correo,...) y formatos. Big Data facilita las operaciones de análisis y seguimiento, favoreciendo la fidelidad y descubrimiento de nuevos mercados.
- Logística: El sector logístico mejora notablemente gracias a las posibilidades analíticas de un Big Data y su potencial para el despliegue de servicios específicos (movilidad, tracking, seguridad, etc.). El ejemplo más popular se encuentra en el control de flotas (la ruta óptima permite a los vehículos circular con la máxima capacidad de carga, pudiendo recorrer rutas mejorando tiempos, consumos y contaminación).

1.6. Base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. [7] Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.

- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

1.6.1. Base de datos relacionales

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base.

Estas bases de datos poseen un conjunto de tablas que contienen datos provistos en categorías predefinidas. Cada tabla (que a veces se llaman ‘relación’) contiene una o más categorías de datos en columnas. Cada fila contiene una instancia única de datos para las categorías definidas por las columnas. [8]

1.6.1.1. Ventajas de las base de datos relacionales

- Está más adaptado su uso y los perfiles que los conocen son mayoritarios y más baratos.
- Debido al largo tiempo que llevan en el mercado, estas herramientas tienen un mayor soporte y mejores suites de productos y add-ons para gestionar estas bases de datos.
- La atomicidad de las operaciones en la base de datos. Esto es, que en estas bases de datos o se hace la operación entera o no se hace utilizando la famosa técnica del rollback.
- Los datos deben cumplir requisitos de integridad tanto en tipo de dato como en compatibilidad.

1.6.1.2. Desventajas de las base de datos relacionales

- La atomicidad de las operaciones juegan un papel crucial en el rendimiento de las bases de datos.
- Escalabilidad, que aunque probada en muchos entornos productivos suele, por norma, ser inferior a las bases de datos NoSQL.

1.6.2. Base de datos no relacionales

Son un enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad), y habitualmente escalan bien horizontalmente.

Son especialmente útil cuando una empresa necesita acceder y analizar grandes cantidades de datos no estructurados o datos que se almacenan de forma remota en varios servidores virtuales en la nube. [9]

1.6.2.1. Ventajas de las base de datos no relacionales

- La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas.
- Suelen ser bases de datos mucho más abiertos y flexibles. Permiten adaptarse a necesidades de proyectos mucho más fácilmente que los modelos de Entidad Relación.
- Se pueden hacer cambios de los esquemas sin tener que parar bases de datos.
- Escalabilidad horizontal: son capaces de crecer en número de máquinas, en lugar de tener que residir en grandes máquinas.
- Se pueden ejecutar en máquinas con pocos recursos.
- Optimización de consultas en base de datos para grandes cantidades de datos.

1.6.2.2. Desventajas de las base de datos no relacionales

- No todas las bases de datos NoSQL contemplan la atomicidad de las instrucciones y la integridad de los datos. Soportan lo que se llama consistencia eventual.
- Problemas de compatibilidad entre instrucciones SQL. Las nuevas bases de datos utilizan sus propias características en el lenguaje de consulta y no son 100 % compatibles con el SQL de las bases de datos relacionales. El soporte a problemas con las queries de trabajo en una base de datos NoSQL es más complicado.
- Falta de estandarización. Hay muchas bases de datos NoSQL y aún no hay un estándar como si lo hay en las bases de datos relacionales. Se presume un futuro incierto en estas bases de datos.
- Soporte multiplataforma. Aún quedan muchas mejoras en algunos sistemas para que soporten sistemas operativos que no sean Linux.
- Suelen tener herramientas de administración no muy usables o se accede por consola.

1.6.2.3. Tipos de base de datos no relacionales

MODELO DE DATOS	FORMATO	CARACTERÍSTICAS	APLICACIONES
Documento.	Similar a JSON (JavaScript Object Notation).	<ul style="list-style-type: none"> - Intuitivo. - Manera natural de modelar datos cercana a la programación orientada a objetos. - Flexibles, con esquemas dinámicos. - Reducen la complejidad de acceso a los datos. 	Se pueden utilizar en diferentes tipos de aplicaciones debido a la flexibilidad que ofrecen.
Grafo.	Nodos con propiedades (atributos) y relaciones (aristas).	<ul style="list-style-type: none"> - Los datos se modelan como un conjunto de relaciones entre elementos específicos. - Flexibles, atributos y longitud de registros variables. - Permite consultas más amplias y jerárquicas. 	Redes sociales, software de recomendación, geolocalización, topologías de red, etc.
Clave-Valor y Columna.	Clave-valor: una clave y su valor correspondiente Columnas: variante que permite más de un valor (columna) por clave.	<ul style="list-style-type: none"> - Rendimiento muy alto. - Alta curva de escalabilidad. - Útil para representar datos no estructurados. 	Aplicaciones que solo utilizan consulta de datos por un solo valor de la clave.

Tabla 1.1: Tipos de Base de Datos NoSql.

1.6.2.3.1. Orientada a columnas

Este tipo de bases de datos están pensadas para realizar consultas y agregaciones sobre grandes cantidades de datos. Funcionan de forma parecida a las bases de datos relacionales, pero almacenando columnas de datos en lugar de registros. Algunos ejemplos de base de datos orientada a columnas: Cassandra, HBase.

- Cassandra: incluida en esta sección, aunque en realidad sigue un modelo híbrido entre orientada a columnas y clave-valor. Es utilizada por Facebook y Twitter (aunque dejaron de usarla para almacenar tweets).

- HBase. Escrita en Java y mantenida por el Proyecto Hadoop de Apache, se utiliza para procesar grandes cantidades de datos. La utilizan Facebook, Twitter o Yahoo.

1.6.2.3.2. Orientada a clave/valor

Son sencillas de entender. Simplemente guardan tuplas que contienen una clave y su valor. Cuando se quiere recuperar un dato, simplemente se busca por su clave y se recupera el valor. Algunos ejemplos de base de datos clave/valor: DynamoDB, Redis.

- DynamoDB: desarrollada por Amazon, es una opción de almacenaje que podemos usar desde los Amazon Web Services. La utilizan el Washington Post y Scopely.
- Redis: desarrollada en C y de código abierto, es utilizada por Craigslist y Stack Overflow (a modo de caché).

1.6.2.3.3. Orientada a documentos

Son aquellas que gestionan datos semi estructurados. Es decir documentos. Estos datos son almacenados en algún formato estándar como puede ser XML, JSON o BSON. Son las bases de datos NoSQL más versátiles. Se pueden utilizar en gran cantidad de proyectos, incluyendo muchos que tradicionalmente funcionarían sobre bases de datos relacionales. Algunos ejemplos de base de datos orientada a documentos: MongoDB, CouchDB.

- MongoDB: probablemente la base de datos NoSQL más famosa del momento. En octubre del año pasado, MongoDB conseguía 150 millones de dólares en financiación, convirtiéndose en una de las startups más prometedoras. Algunas compañías que actualmente utilizan MongoDB son Foursquare o eBay.
- CouchDB: es la base de datos orientada a documentos de Apache. Una de sus interesantes características es que los datos son accesibles a través de una API Rest. Este sistema es utilizado por compañías como Credit Suisse y la BBC.

1.6.2.3.4. Orientada a grafo

Basadas en la teoría de grafos utilizan nodos y aristas para representar los datos almacenados. Son muy útiles para guardar información en modelos con muchas relaciones, como redes y conexiones sociales. Algunos ejemplos de base de datos orientada a grafos: Infinite Graph, Neo4j.

- Infinite Graph: escrita en Java y C++ por la compañía Objectivity. Tiene dos modelos de licenciamiento: uno gratuito y otro de pago.

- Neo4j: base de datos de código abierto, escrita en Java por la compañía Neo Technology. Utilizada por compañías como HP, Infojobs o Cisco.

1.7. Almacén de datos (Data Warehouse)

Un almacén de datos es una base de datos corporativa o repositorio de datos, que se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde infinitas perspectivas y con grandes velocidades de respuesta. La creación de un almacén de datos colecciona datos orientado a temas, integrado, no volátil, de tiempo variante el cual representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de inteligencia de negocios. Existen 2 investigadores muy famosos, Bill Inmon y Ralph Kimball relacionados con este concepto de almacén de datos. [10]

Para Bill Inmon (1992), quien acuñó el término por primera vez, “el Almacén de Datos es una colección de datos orientados al tema, integrados, no volátiles e históricos, organizados para el apoyo de un proceso de ayuda a la decisión. No obstante, y como cabe suponer, es mucho más que eso”.

Para Ralph Kimball (1997) “el Almacén de Datos es una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis; es la unión de todos las Bodegas de Datos de una entidad”.

1.8. Inteligencia de negocios

Inteligencia de negocios es el conjunto de conceptos y métodos para mejorar la toma de decisiones en los negocios, utilizando sistemas de apoyo basados en hechos (Howard Dresner, 1989). Sin embargo, en la actualidad este concepto incluye una amplia categoría de metodologías, aplicaciones y tecnologías que permiten reunir, acceder, transformar y analizar los datos con el propósito de ayudar a los usuarios de una organización a tomar mejores decisiones de negocio [11]

1.9. Lenguajes de programación

En computación, un lenguaje de programación es cualquier lenguaje artificial ya que intenta conservar una similitud con el lenguaje humano, el cual, se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. [12]

Establecen un conjunto de símbolos, reglas sintácticas y semánticas, las cuales rigen la estructura y el significado del programa, junto con sus elementos y expresiones. De esta forma, permiten a los programadores o desarrolladores,

poder especificar de forma precisa los datos sobre los que se va a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas. Usualmente se clasifican en interpretados y compilados, en el cual los compilados tienen un compilador específico que obtiene como entrada un programa y traduce las instrucciones las cuales pueden servir de entrada para otro interprete o compilado y los interpretados tienen un intérprete específico que obtiene como entrada un programa y ejecuta las acciones escritas a medida que las va procesando.

1.9.1. R

R es un lenguaje y entorno de programación para análisis estadístico y gráfico, el cual proporciona un amplio abanico de herramientas estadísticas (modelos lineales y no lineales, tests estadísticos, análisis de series temporales, algoritmos de clasificación y agrupamiento, etc.) y gráficas. Se trata de un proyecto de software libre, resultado de la implementación GNU del premiado lenguaje S y se distribuye bajo la licencia GNU GPL y está disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux. Puede integrarse con distintas bases de datos y existen bibliotecas que facilitan su utilización desde lenguajes de programación interpretados como Perl y Python. [13]

R al estar orientado a las estadísticas, proporciona un amplio abanico de herramientas de cálculo numérico y a su vez para minería de datos, y posee una capacidad gráfica, que permite generar gráficos con alta calidad, con sólo utilizar las funciones de graficación.

1.9.1.1. R STUDIO

RStudio es un entorno de desarrollo integrado (IDE) construido exclusivo para R, para computación estadística y gráficos . Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo. [14]

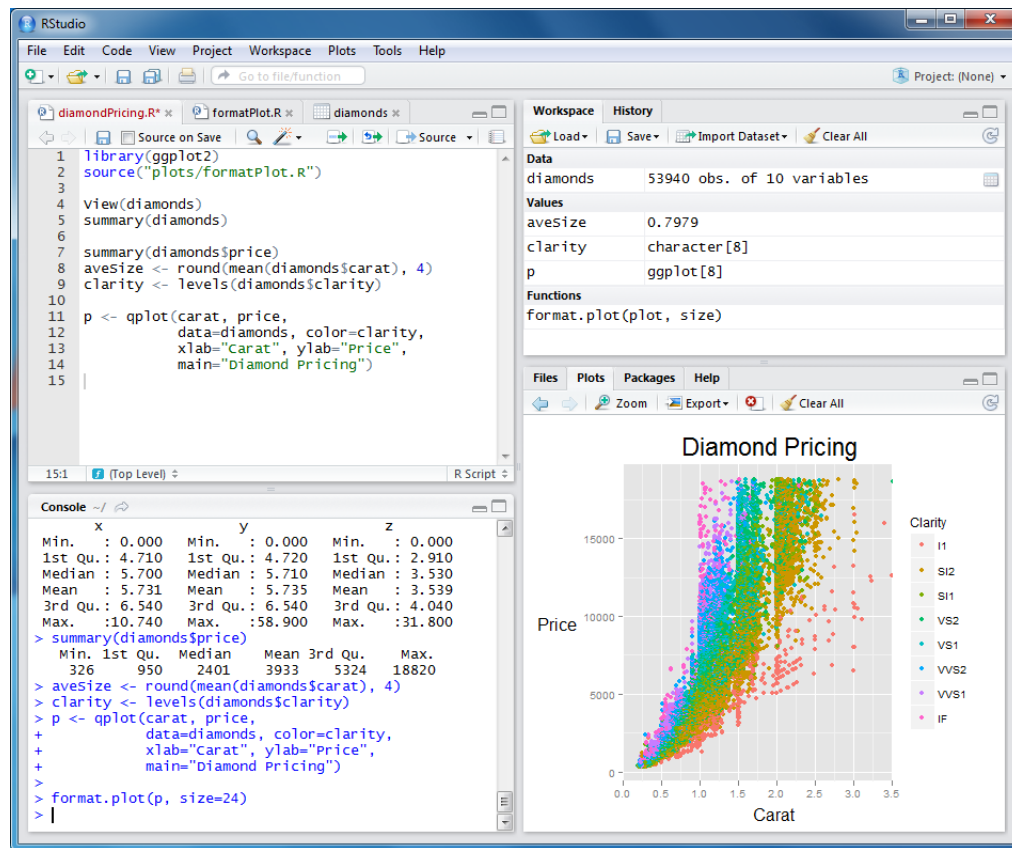


Figura 1.2: IDE RSTUDIO.

Algunas de sus características son:

- Ejecutar código R directamente desde el editor de código fuente.
- Salto rápido a las funciones definidas.
- Colaborativo.
- Documentación y soporte integrado.
- Administración sencilla de múltiples directorios de trabajo mediante proyectos.
- Navegación en espacios de trabajo y visor de datos.
- Potente autoría y depuración.
- Depurador interactivo para diagnosticar y corregir los errores rápidamente.

- Herramientas de desarrollo extensas.
- Autoría con Sweave y R Markdown.

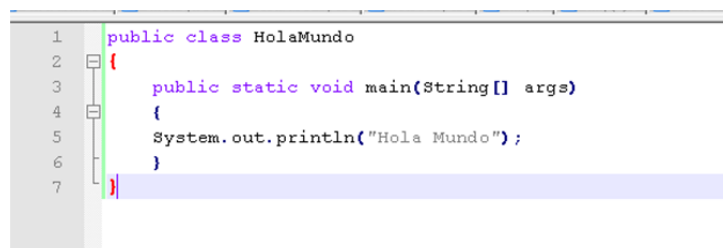
1.9.2. JAVA

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. [15]

El lenguaje Java se creó con cinco objetivos principales:

- Debería incluir por defecto soporte para trabajo en red.
- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.

A screenshot of a code editor with a light blue background. The code is written in Java and is as follows:

```
1 public class HolaMundo
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hola Mundo");
6     }
7 }
```

The code is color-coded: keywords like 'public', 'class', 'static', 'void', and 'String' are in purple; 'System.out' is in blue; and the string literal is in red. The line numbers 1 through 7 are on the left side of the editor.

Figura 1.3: Hola mundo en Java.

1.9.3. PYTHON

Python es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma, cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License. [16] Algunas de las características de Python son:

- Orientado a Objetos : La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- Funciones y librerías: Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- Propósito general: Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
- Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- Interpretado: Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- Interactivo: Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- Sintaxis clara: Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

1.9.4. JAVASCRIPT

Es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Es ampliamente utilizado en el mundo del desarrollo web por ser muy versátil y potente, tanto del lado del cliente y del servidor. [17] Algunas de las cosas que se pueden desarrollar con javascript son:

- Páginas dinámicas (DHTML).
- Comprobación de datos (Formularios).
- Uso de los elementos de la página web.
- Intercambiar información entre páginas web en distintas ventanas.
- Manipulación de gráficos, texto, etc...
- Comunicación con plug-ins: Flash, Java, Shockwave, etc...

1.9.4.1. NODE.JS

Node.js es un entorno o interfaz en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. Node.js trabaja con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deba realizar. [18]

1.9.4.2. ANGULAR.JS

AngularJS es un framework JavaScript de desarrollo de aplicaciones web en el lado cliente, viene de la mano de los desarrolladores de Google y se podría decir que utiliza el patrón MVC (Model-View-Controller), aunque ellos mismos lo definen más bien como un MVW (Model-View-Whatever (whatever works for you)). Los creadores de este framework están convencidos de que HTML no está aún preparado para servir vistas dinámicas de un modo eficiente, así que han decidido extender la sintaxis de HTML para darle más funcionalidad.[19]

1.9.4.3. D3.JS

D3.js es una biblioteca de JavaScript para la manipulación de documentos basados en datos. D3 ayuda a llevar los datos a la vida usando HTML, SVG y CSS. Permite crear visualizaciones complejas y gráficos interactivos. La librería está expandiendo los límites de las visualizaciones que pretenden contar historias con datos. La librería permite manipular documentos basados en datos usando estándares abiertos de la web y los navegadores pueden crear visualizaciones complejas sin depender de un software propietario. [20]

1.10. APACHE HADOOP

En el siguiente capítulo se mencionaran y explicará brevemente la definición de Apache Hadoop y sus componentes.

1.10.1. Definición de APACHE HADOOP

Apache Hadoop es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre de la comunidad Apache. Permite el procesamiento de grandes volúmenes de datos de forma distribuida a través de clústeres usando modelos sencillos de programación. Está siendo construido y usado por una comunidad global de contribuyentes, mediante el lenguaje de programación Java. Está diseñado para escalar desde un servidor sencillo hasta miles de nodos los cuales pueden ser heterogéneos. [21]

1.10.1.1. COMMONS

Apache Commons es un conjunto de proyectos de Apache Software Foundation, que originalmente formaron parte de Jakarta Project. El propósito de estos proyectos consiste en proveer componentes de software Java reutilizables, en código abierto. [22] El proyecto Apache Commons se compone de tres partes:

- Commons Proper: Un repositorio de componentes Java reutilizables.
- Commons Sandbox: Un espacio de trabajo para el desarrollo de componentes de Java.
- Commons Dormant: Un repositorio de componentes que se encuentran actualmente inactivas.

1.10.1.2. MAPREDUCE

Hadoop MapReduce es un marco de software o un modelo de programación para el procesamiento distribuido de grandes conjuntos de datos en clústeres de cómputo de hardware de los productos básicos, que tiene como objetivo principal dividir los datos de entrada en partes independientes que son procesados de forma totalmente paralela. Es un subproyecto del proyecto Hadoop. Este es el paradigma de programación que permite la escalabilidad masiva a través de cientos o miles de servidores en un cluster Hadoop. [23]

El Framework MapReduce tiene una arquitectura maestro / esclavo. Cuenta con un servidor maestro o JobTracker y varios servidores esclavos o TaskTrackers, uno por cada nodo del clúster. El JobTracker es el punto de interacción entre los usuarios y el framework MapReduce. Los usuarios envían trabajos MapReduce al JobTracker, que los pone en una cola de trabajos pendientes y los ejecuta en el orden de llegada.

El JobTracker gestiona la asignación de tareas y delega las tareas a los TaskTrackers. Los TaskTrackers ejecutan tareas bajo la orden del JobTracker y también manejan el movimiento de datos entre la fase Map y Reduce. Para ver las diferencias entre JobTracker y TaskTracker vamos a ver las características de cada uno.

1.10.1.2.1. JobTracker

- Capacidad para manejar metadatos de trabajos.
- Estado de la petición del trabajo.
- Estado de las tareas que se ejecutan en TaskTracker.
- Decide sobre la programación.
- Hay exactamente un JobTracker por cluster.
- Recibe peticiones de tareas enviadas por el cliente.
- Programa y monitoriza los trabajos MapReduce con TaskTrackers.

1.10.1.2.2. TaskTracker

- Ejecuta las solicitudes de trabajo de JobTrackers.
- Obtiene el código que se ejecutará.
- Aplica la configuración específica del trabajo.
- Comunicación con el JobTracker: Envíos de la salida, finalizar tareas, actualización de tareas, etc.

1.10.1.2.3. Map

La función Map recibe como parámetros un par de (clave, valor) y devuelve una lista de pares. Esta función se encarga del mapeo y se aplica a cada elemento de la entrada de datos, por lo que se obtendrá una lista de pares por cada llamada a la función Map. Después se agrupan todos los pares con la misma clave de todas las listas, creando un grupo por cada una de las diferentes claves generadas. No hay requisito de que el tipo de datos para la entrada coincida con la salida y no es necesario que las claves de salida sean únicas.

1.10.1.2.4. Reduce

La función Reduce se aplica en paralelo para cada grupo creado por la función Map(). La función Reduce se llama una vez para cada clave única de la salida de la función Map. Junto con esta clave, se pasa una lista de todos los valores asociados con la clave para que pueda realizar alguna fusión para producir un conjunto más pequeño de los valores.

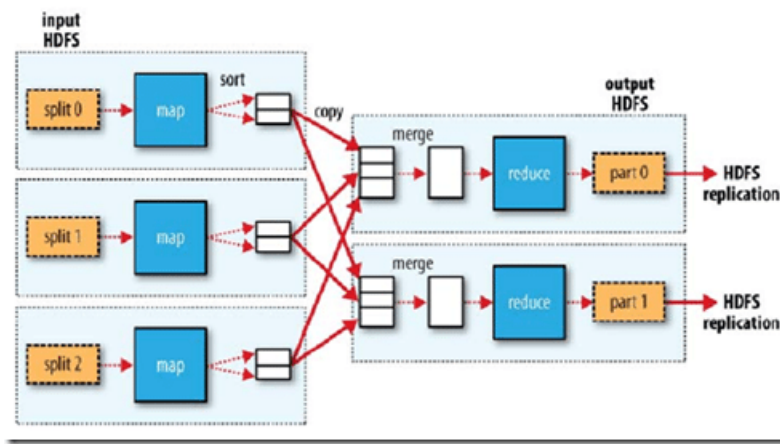


Figura 1.4: Funcionamiento de MapReduce.

Cuando se inicia la tarea Reduce, la entrada se encuentra dispersa en varios archivos a través de los nodos en las tareas de Map. Los datos obtenidos de la fase Map se ordenan para que los pares clave-valor sean contiguos (fase de ordenación, sort fase), esto hace que la operación Reduce se simplifique ya que el archivo se lee secuencialmente.

Si se ejecuta el modo distribuido estos necesitan ser primero copiados al filesystem local en la fase de copia. Una vez que todos los datos están disponibles a nivel local se adjuntan a una fase de adición, el archivo se fusiona (merge) de forma ordenado. Al final, la salida consistirá en un archivo de salida por tarea reduce ejecutada.

Por lo tanto, N archivos de entrada generará M mapas de tareas para ser ejecutados y cada mapa de tareas generará tantos archivos de salida como tareas Reduce hayan configuradas en el sistema.

1.10.1.3. HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

El Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido, escalable y portátil escrito en Java para el framework Hadoop. Cada

nodo en una instancia Hadoop típicamente tiene un único nodo de datos; un clúster de datos forma el clúster HDFS. La situación es típica porque cada nodo no requiere un nodo de datos para estar presente. Cada nodo sirve bloques de datos sobre la red usando un protocolo de bloqueo específico para HDFS.

El sistema de archivos usa la capa TCP/IP para la comunicación; los clientes usan RPC para comunicarse entre ellos. El HDFS almacena archivos grandes, a través de múltiples máquinas. Consigue fiabilidad mediante replicado de datos a través de múltiples hosts, y no requiere almacenamiento RAID en ellos. Con el valor de replicación por defecto, 3, los datos se almacenan en 3 nodos: dos en el mismo rack, y otro en un rack distinto. Los nodos de datos pueden hablar entre ellos para reequilibrar datos, mover copias, y conservar alta la replicación de datos. HDFS no cumple totalmente con POSIX porque los requerimientos de un sistema de archivos POSIX difieren de los objetivos de una aplicación Hadoop, porque el objetivo no es tanto cumplir los estándares POSIX sino la máxima eficacia y rendimiento de datos. HDFS fue diseñado para gestionar archivos muy grandes. [24]

1.10.1.3.1. Características de HDFS

- Es adecuado para el almacenamiento y procesamiento distribuido.
- Hadoop proporciona una interfaz de comandos para interactuar con HDFS.
- Los servidores de namenode datanode y ayudan a los usuarios a comprobar fácilmente el estado del clúster.
- Streaming el acceso a los datos del sistema de ficheros.
- HDFS proporciona permisos de archivo y la autenticación.

1.10.1.3.2. Arquitectura de HDFS

HDFS es un sistema de ficheros pensado para el almacenamiento de ficheros “grandes” (por encima de 100 MB) y en la que el acceso a esa información está orientado hacia procesamiento en batch o lectura de tipo “write once”-“read-many-times”. En un cluster de HDFS encontramos dos tipos de nodos diferentes y al cliente:

- Namenodes: son los encargados de gestionar el espacio de nombres del sistema de ficheros. Se encarga de la administración del sistema de archivos, mantiene el sistema de archivos y la metadata asociada a estos y a los directorios. Almacena los datos relacionados con la posición de un bloque para un archivo en específico durante un periodo de tiempo, estos datos son actualizados al iniciar el sistema y cada cierto tiempo.

- Datanodes: son los que almacenan los bloques de información y los recuperan bajo demanda. Se encargan de almacenar y distribuir los bloques entre los distintos nodos. La distribución de los bloques es realizada cuando reciben un aviso desde un Namenode o algún cliente solicita una distribución. Una vez realizada la distribución, estos realizan un reporte al Namenode, este reporte se realiza periódicamente y contiene los bloques los cuales están siendo almacenados en ese nodo en específico.
- Clientes: son los que acceden al sistema de archivos.

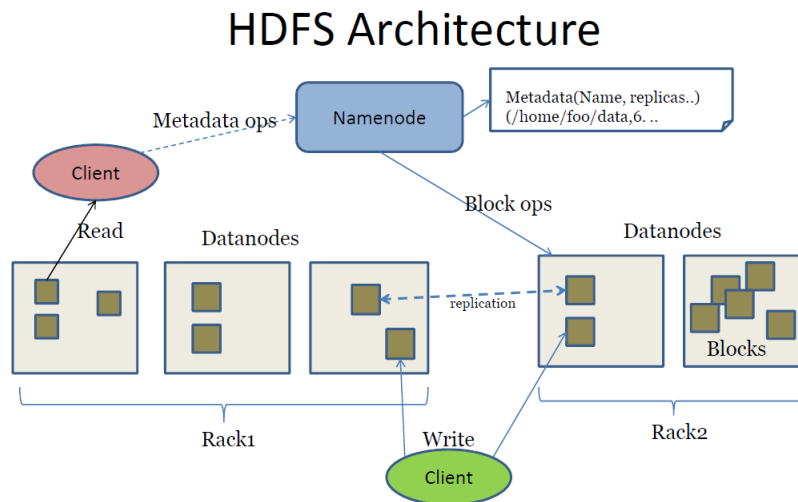


Figura 1.5: Arquitectura HDFS.

1.10.1.4. Yet Another Resource Negotiator (YARN)

Apache Hadoop YARN (por las siglas en inglés de “otro negociador de recursos”) es una tecnología de administración de clústeres. YARN es una de las características clave de la segunda generación de la versión Hadoop 2 del marco de procesamiento distribuido de código abierto de Apache Software Foundation. Originalmente descrito por Apache como un gestor de recursos rediseñado, YARN se caracteriza ahora como un sistema operativo distribuido, a gran escala, para aplicaciones de Big Data.

YARN combina un administrador central de recursos que reconcilia la forma en que las aplicaciones utilizan los recursos del sistema de Hadoop con los agentes de administración de nodo que monitorean las operaciones de procesamiento de nodos individuales del clúster. Ejecutándose en clústeres de hardware

básicos, Hadoop ha atraído un interés particular como zona de espera y de almacenamiento de datos para grandes volúmenes de datos estructurados y no estructurados destinados al uso en aplicaciones de analítica. Separar HDFS de MapReduce con YARN hace al ambiente Hadoop más adecuado para las aplicaciones operativas que no pueden esperar para que terminen los trabajos por lotes. [25]

1.10.1.4.1. Arquitectura de YARN

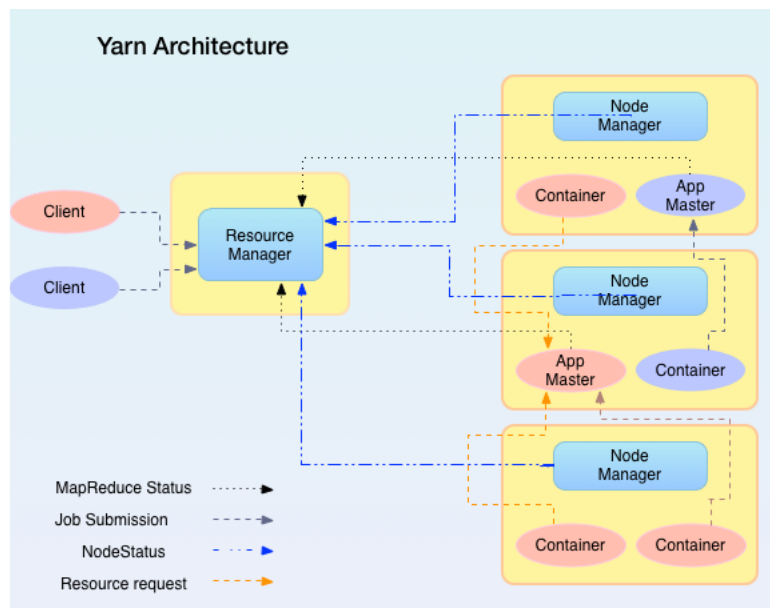


Figura 1.6: Arquitectura Yarn.

El ResourceManager y el NodeManager (NM) esclavo de cada nodo forman el entorno de trabajo, encargándose el ResourceManager de repartir y gestionar los recursos entre todas las aplicaciones del sistema mientras que el ApplicationMaster se encarga de la negociación de recursos con el ResourceManager y los NodeManager para poder ejecutar y controlar las tareas, es decir, les solicita recursos para poder trabajar.

1.10.2. Herramientas del ecosistema

El ecosistema de Hadoop posee un conjunto de herramientas, las cuales a su vez poseen subconjuntos, en este capítulo mencionaremos algunos de ellos y una breve descripción.

1.10.2.1. Administración de los datos

Son herramientas que permiten el almacenamiento y el manejo de datos, como HDFS y YARN.

1.10.2.2. Acceso a los datos

Las herramientas que se encargan del acceso a los datos a los usuarios del ecosistema Hadoop, permiten la lectura y también el almacenamiento de los datos.

1.10.2.2.1. APACHE SPARK

Spark es una plataforma de computación de código abierto para análisis y procesos avanzados, que tiene muchas ventajas sobre Hadoop. Desde el principio, Spark fue diseñado para soportar en memoria algoritmos iterativos que se pudiesen desarrollar sin escribir un conjunto de resultados cada vez que se procesaba un dato. Esta habilidad para mantener todo en memoria es una técnica de computación de alto rendimiento aplicado al análisis avanzado, la cual permite que Spark tenga unas velocidades de procesamiento que sean 100 veces más rápidas que las conseguidas utilizando MapReduce.

Spark tiene un framework integrado para implementar análisis avanzados que incluye la librería MLlib, el motor gráfico GraphX, Spark Streaming, y la herramienta de consulta Shark. Esta plataforma asegura a los usuarios la consistencia en los resultados a través de distintos tipos de análisis. [26]

Spark mantiene la escalabilidad lineal y la tolerancia a fallos de MapReduce, pero amplía sus bondades gracias a varias funcionalidades: DAG y RDD.

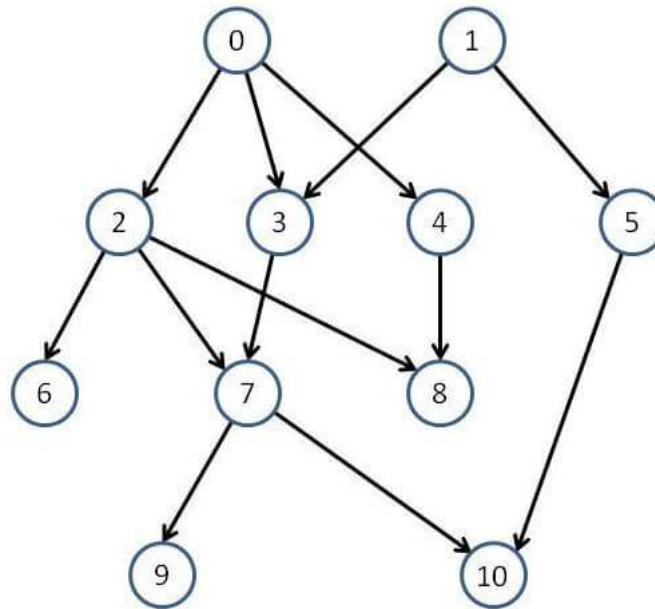


Figura 1.7: DAG (Directed Acyclic Graph).

- DAG (Grafo Acíclico Dirigido): Es un grafo dirigido que no tiene ciclos, es decir, para cada nodo del grafo no hay un camino directo que comience y finalice en dicho nodo. Un vértice se conecta a otro, pero nunca a si mismo.

Spark soporta el flujo de datos acíclico. Cada tarea de Spark crea un DAG de etapas de trabajo para que se ejecuten en un determinado cluster. En comparación con MapReduce, el cual crea un DAG con dos estados predefinidos (Map y Reduce), los grafos DAG creados por Spark pueden tener cualquier número de etapas. Spark con DAG es más rápido que MapReduce por el hecho de que no tiene que escribir en disco los resultados obtenidos en las etapas intermedias del grafo. MapReduce, sin embargo, debe escribir en disco los resultados entre las etapas Map y Reduce. Gracias a una completa API, es posible programar complejos hilos de ejecución paralelos en unas pocas líneas de código.

- RDD (Resilient Distributed Dataset): Apache Spark mejora con respecto a los demás sistemas en cuanto a la computación en memoria. RDD permite a los programadores realizar operaciones sobre grandes cantidades de datos en clusters de una manera rápida y tolerante a fallos. Surge debido a que las herramientas existentes tienen problemas que hacen que se manejen los datos ineficientemente a la hora de ejecutar algoritmos iterativos y procesos de minería de datos. En ambos casos, mantener los datos en memoria puede mejorar el rendimiento considerablemente.

Una vez que los datos han sido leídos como objetos RDD en Spark, pueden realizarse diversas operaciones mediante sus APIs. Los dos tipos de operaciones que se pueden realizar son:

- Transformaciones: tras aplicar una transformación, obtenemos un nuevo y modificado RDD basado en el original.
- Acciones: una acción consiste simplemente en aplicar una operación sobre un RDD y obtener un valor como resultado, que dependerá del tipo de operación.

Dado que las tareas de Spark pueden necesitar realizar diversas acciones o transformaciones sobre un conjunto de datos en particular, es altamente recomendable y beneficioso en cuanto a eficiencia el almacenar RDDs en memoria para un rápido acceso a los mismos. Mediante la función `cache()` se almacenan los datos en memoria para que no sea necesario acceder a ellos en disco.

El almacenamiento de los datos en memoria caché hace que los algoritmos de machine learning ejecutados que realizan varias iteraciones sobre el conjunto de datos de entrenamiento sea más eficiente. Además, se pueden almacenar versiones transformadas de dichos datos.

■ Modelo de programación:

Un programa típico se organiza de la siguiente manera:

1. A partir de una variable de entorno llamada `context` se crea un objeto RDD leyendo datos de fichero, bases de datos o cualquier otra fuente de información.
2. Una vez creado el RDD inicial se realizan transformaciones para crear más objetos RDD a partir del primero. Dichas transformaciones se expresan en términos de programación funcional y no eliminan el RDD original, sino que crean uno nuevo.
3. Tras realizar las acciones y transformaciones necesarias sobre los datos, los objetos RDD deben converger para crear el RDD final. Este RDD puede ser almacenado.

Cuando el programa comienza su ejecución crea un grafo similar al de la figura siguiente en el que los nodos son objetos RDD y las uniones entre ellos son operaciones de transformación. El grafo de la ejecución es un DAG y, cada grafo es una unidad atómica de ejecución. En la figura siguiente, las líneas rojas representan transformación y las verdes operación.

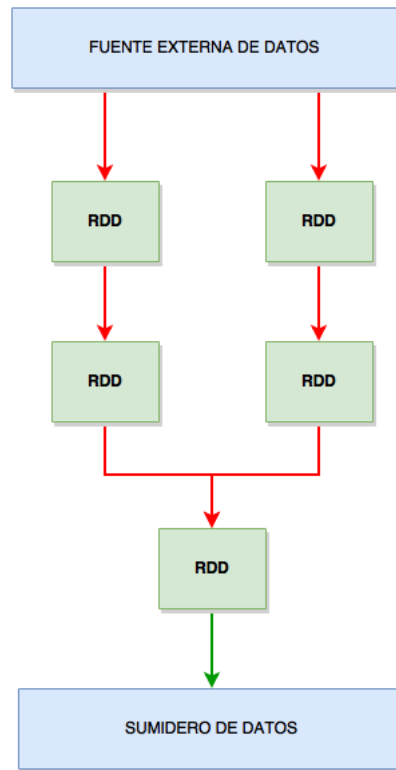


Figura 1.8: Proceso Resilient Distributed Dataset.

■ Tipos de transformaciones:

Es muy posible que los datos con los que se necesite tratar estén en diferentes objetos RDD, por lo que Spark define dos tipos de operaciones de transformación: narrow transformation y wide transformation.

- Narrow transformation: se utiliza cuando los datos que se necesitan tratar están en la misma partición del RDD y no es necesario realizar una mezcla de dichos datos para obtenerlos todos. Algunos ejemplos son las funciones `filter()`, `sample()`, `map()` o `flatMap()`.
- Wide transformation: se utiliza cuando la lógica de la aplicación necesita datos que se encuentran en diferentes particiones de un RDD y es necesario mezclar dichas particiones para agrupar los datos necesarios en un RDD determinado. Ejemplos de wide transformation son: `groupByKey()` o `reduceByKey()`.

Una representación gráfica de ambos tipos de transformaciones es la que se puede apreciar en la figura siguiente:

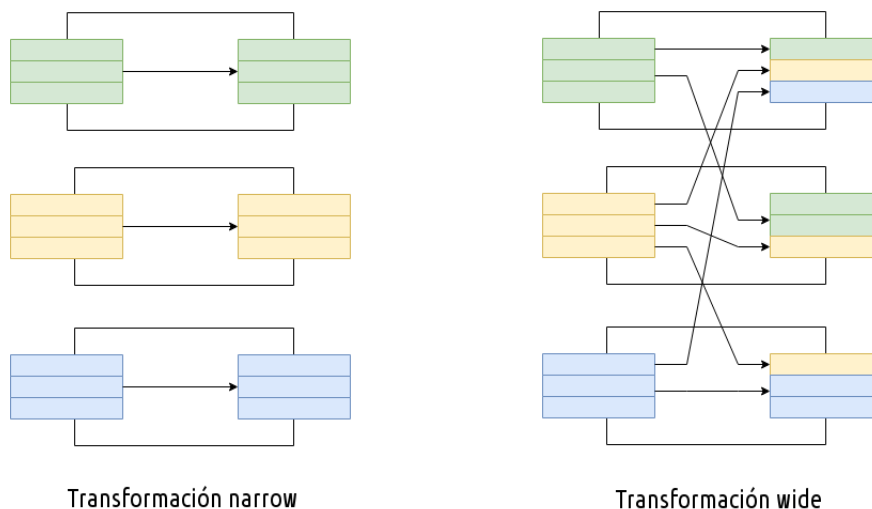


Figura 1.9: Narrow transformation y Wide transformation.

En algunos casos es posible realizar un reordenamiento de datos para reducir la cantidad de datos que deben ser mezclados. A continuación se muestra un ejemplo de un JOIN entre dos objetos RDD seguido de una operación de filtrado.

■ Ventajas de Spark:

- Spark tiene más potencia que hadoop:

Para empezar Spark es un framework de análisis distribuido en memoria y nos permite ir más allá de las operaciones en batch de Hadoop MapReduce: procesamiento de streaming, machine learning (MLlib), cálculo de grafos (GraphX), integración con lenguaje R (Spark R) y análisis interactivos.

Con todo esto, ahora se puede desarrollar nuevos proyectos de big data con menos presupuesto y soluciones más completas.

- Spark es rápido, muy rápido:

Spark puede ejecutar análisis de varios órdenes de magnitud más rápido que los despliegues de Hadoop existentes. Esto significa una mayor interactividad, la experimentación más rápido y mayor productividad para los analistas.

- Spark puede coexistir con tu arquitectura de Big Data:

Puede coexistir con las instalaciones existentes de Hadoop y añadir nuevas funcionalidades. Spark se integra perfectamente con Hadoop

y en muchos de los proyectos se utiliza/almacenan los datos que están en el sistema de fichero de Hadoop HDFS y/o se ejecutan los procesos de Spark usando YARN de Hadoop 2.0. Además puede funcionar con muchos otros productos de Big Data como: CassandraDB, Google Big Query, almacenamiento de Amazon S3, Elastic Search, etc.

- Spark entiende SQL:

El módulo Spark Sql es capaz de usar fuentes de datos existentes (HIVE, CassandraDB, MongoDB, JDBC, etc), se puede usar para gestionar las fuentes internas de datos (RDDs - DataFrames) como fueran tablas estructurados, y que las inversiones realizadas en herramientas de BI se puedan acceder a la información gestionada por Spark. Aunque Spark SQL no es la implementación más robusta y completa del mercado ya está lista para ser usada.

- Spark mima a los desarrolladores:

Cuando una tecnología encanta a los desarrolladores se convierten en early adopters y empiezan a usarla y disfrutarla. Spark es un ejemplo de esto, cuando usan Spark solo tiene que dedicarse a resolver el problema. Spark se ha programado con el lenguaje Scala que un nuevo lenguaje funcional y orientado a objetos. Gracias a Scala son capaces de programar de manera muy concisa y fluida soluciones que antes requerían cientos de líneas. Además se puede programar en python, R e incluso en Java.

- Spark empieza a ser el motor de Big Data:

Ahora mismo Apache Spark forma muchos proyectos de Big Data y empresas como IBM, Microsoft, Amazon, Google lo integran con sus productos de Big Data.

1.10.2.2.2. APACHE ACCUMULO

Apache Accumulo es un store clave/valor distribuido, escalable y de alto rendimiento. Se basa en el diseño de Google BigTable y se construye sobre Hadoop, Zookeeper y Thrift.

Accumulo permite el manejo de datos a nivel de celdas, lo cual es una funcionalidad muy importante debido a que se puede restringir el acceso a los datos a ciertos usuarios en específico. Permite también la mezcla de distintos datos los cuales pueden estar restringidos o no, las reglas que pueden ser aplicadas a los datos pueden llegar a ser muy específicas. [27]

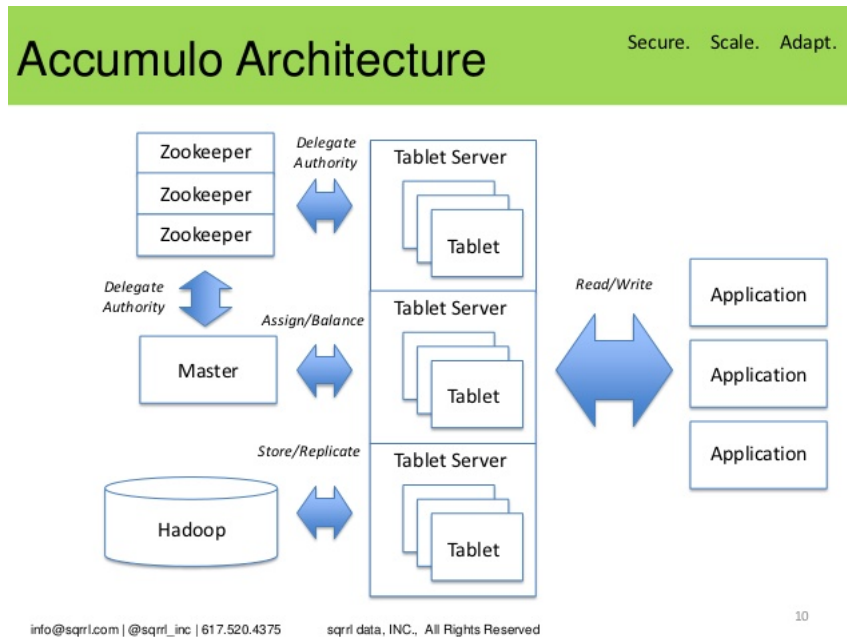


Figura 1.10: Arquitectura Apache Accumulo.

1.10.2.2.3. APACHE HBASE

HBase, se trata de la base de datos de Hadoop. HBase es el componente de Hadoop a usar, cuando se requiere escrituras/lecturas en tiempo real y acceso aleatorio para grandes conjuntos de datos. Es una base de datos orientada a la columna, eso quiere decir que no sigue el esquema relacional. No admite SQL. [28]

Características de HBase:

- HBase es lineal escalable.
- Ha hecho automático.
- Proporciona lectura coherente y escrituras.
- Se integra con Hadoop, tanto como un origen y un destino.
- Tiene fácil API de java para el cliente.
- Proporciona replicación de datos en clústeres.

Las aplicaciones de HBase:

- Apache HBase se utiliza para tener al azar y en tiempo real de acceso de lectura/escritura a los grandes datos.

- Alberga las tablas de gran tamaño en la parte superior de los grupos de hardware de productos básicos.
- Apache HBase es una base de datos relacional basada en Bigtable de Google. Actos de Bigtable de Google File System, del mismo modo Apache HBase trabaja en la parte superior de Hadoop y HDFS.
- Se usa cuando es necesario escribir aplicaciones pesadas.
- HBase se utiliza cada vez que necesitemos para proporcionar un rápido acceso aleatorio a los datos disponibles.
- Empresas como Facebook, Twitter, Yahoo y Adobe uso HBase internamente.

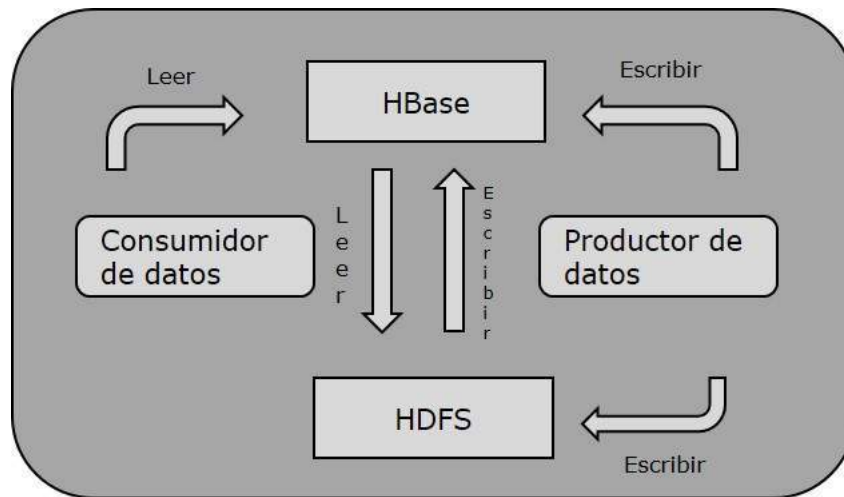


Figura 1.11: Modelo utilizando Apache HBase.

Mecanismo de almacenamiento en HBase:

HBase es una columna de base de datos y las tablas en que se ordenan por fila. Esquema de la tabla define solamente la columna las familias, que son los pares de valor clave. Una tabla con varias columnas y cada columna familias familia puede tener cualquier número de columnas. Los valores de columna se almacenan físicamente en el disco. Cada valor de la celda de la tabla tiene una marca de tiempo. En resumen, en un HBase:

- Tabla es un conjunto de filas.
- Fila es una colección de la columna.
- Columna familia es una colección de columnas.
- Columna es una recopilación de los principales pares de valores.

HDFS	HBase
HDFS es un sistema de ficheros distribuido adecuado para almacenar archivos de gran tamaño.	HBase es una base de datos creada en la parte superior de la HDFS.
HDFS no admite búsquedas rápidas registro individual.	HBase proporciona búsquedas rápidas tablas más grandes.
Proporciona una alta latencia procesamiento por lotes; un concepto de procesamiento por lotes.	Proporciona acceso de baja latencia a filas de miles de millones de registros (acceso aleatorio).
Sólo proporciona acceso secuencial de los datos.	HBase internamente usa tablas Hash y proporciona acceso aleatorio, y que almacena los datos en archivos indexados HDFS búsquedas más rápido.

Tabla 1.2: Comparación entre HBase y HDFS.

1.10.2.2.4. APACHE HIVE

Hive es un sistema de Data Warehouse para Hadoop que facilita el uso de la agregación de los datos, ad-hoc queries, y el análisis de grandes datasets almacenados en Hadoop. Hive proporciona métodos de consulta de los datos usando un lenguaje parecido al SQL, llamado HiveQL. Además permite de usar los tradicionales MapReduce cuando el rendimiento no es el correcto. Tiene interfaces JDBC/ODBC, por lo que empieza a funcionar su integración con herramientas de inteligencia de negocios. [29]

Hive Architecture

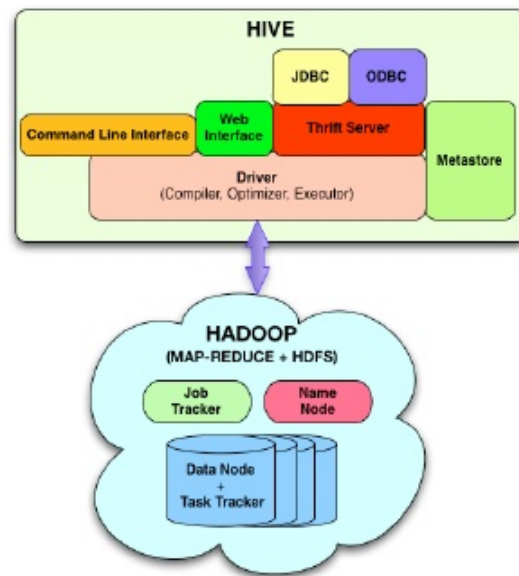


Figura 1.12: Arquitectura Apache Hive.

Características de Hive:

- Esquema que almacena en una base de datos y se procesan los datos en HDFS.
- Está diseñado para OLAP.
- Proporciona tipo SQL lenguaje de consulta o pedido HiveQL HQL.
- Es familiar, rápido, escalable y extensible.

1.10.2.2.5. APACHE STORM

Apache Storm es un sistema que sirve para recuperar streams de datos en tiempo real desde múltiples fuentes de manera distribuida, tolerante a fallos y en alta disponibilidad. Storm está principalmente pensado para trabajar con datos que deben ser analizados en tiempo real, por ejemplo datos de sensores que se emiten con una alta frecuencia o datos que provengan de las redes sociales donde

a veces es importante saber qué se está compartiendo en este momento.

Se compone de dos partes principalmente. La primera es la que se denomina Spout y es la encargada de recoger el flujo de datos de entrada. La segunda se denomina Bolt y es la encargada del procesado o transformación de los datos. [30]

1.10.2.2.6. APACHE MAHOUT

Es un proyecto de la Fundación Apache Software para producir gratuitas implementaciones distribuidas o no escalables de aprendizaje automático algoritmos se centraron principalmente en las áreas de colaboración filtrado, agrupación y clasificación. Muchas de las implementaciones utilizan el Hadoop plataforma. Mahout también proporciona bibliotecas de Java para las operaciones matemáticas comunes (centrado en álgebra lineal y estadística) y las colecciones de Java primitivos. [31]

Mahout soporta cuatro casos principales de uso de la ciencia de datos:

- Colaboración de filtrado: Comportamiento del usuario y hace recomendaciones de productos (por ejemplo, recomendaciones de Amazon).
- Clustering: Toma elementos de una clase en particular (como páginas web o artículos de prensa) y los organiza en grupos de origen natural, de tal manera que los elementos pertenecientes a un mismo grupo son similares entre sí.
- Clasificación: Aprende de categorizaciones existentes y luego asigna artículos no clasificados en la categoría del mejor.
- Minería de elementos frecuentes: Análisis de elementos de un grupo (por ejemplo, artículos en un carro de compras o términos en una sesión de consulta) y luego identifica los elementos que suelen aparecer juntos.

1.10.2.3. Integración

Aquellas herramientas que facilitan la extracción, transformación, replicación, entre otros, de los datos, son las herramientas de integración.

1.10.2.3.1. APACHE FALCON

Falcon es un sistema de procesamiento y manejo de carga de datos destinada a facilitar a los consumidores finales a bordo de sus procesamiento de datos. Establece relación entre los diversos elementos de procesamiento de datos y en un entorno de Hadoop. Posee soporte para el manejo de datos finales, integración con MetaStore/Hive/HCatalog. Proporciona notificaciones al cliente final sobre

la base de la disponibilidad de los grupos de carga de datos.

Básicamente, es una herramienta de software que simplifica la creación y el manejo de tuberías (pipelines) de procesamiento de datos. [32]

1.10.2.3.2. APACHE FLUME

Apache Flume es un producto que forma parte del ecosistema Hadoop, y conforma una solución Java distribuida y de alta disponibilidad para recolectar, agregar y mover grandes cantidades de datos desde diferentes fuentes a un data store centralizado.

Surge para subir datos de aplicaciones al HDFS de Hadoop. Su Arquitectura se basa en flujos de streaming de datos, ofrece mecanismos para asegurar la entrega y mecanismos de failover y recuperación. Ofrece una gestión centralizada. [33]

1.10.2.3.3. APACHE SQOOP

Apache Sqoop (“Sql-to-Hadoop”), es una herramienta diseñada para transferir de forma eficiente bulk data entre Hadoop y sistemas de almacenamiento con datos estructurados, como bases de datos relacionales [34]. Algunas de sus características son:

- Permite importar tablas individuales o bases de datos enteras a HDFS.
- Genera clases Java que permiten interactuar con los datos importados.
- Además, permite importar de las bases de datos SQL a Hive.

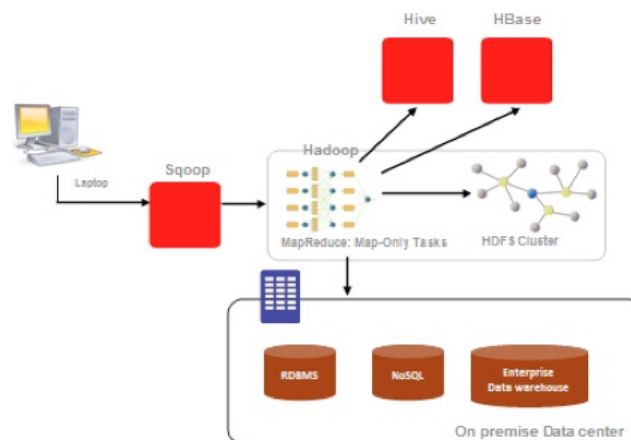


Figura 1.13: Sqoop.

1.10.2.4. Seguridad

Apache Hadoop provee herramientas de monitoreo y administración que aportan seguridad a los datos, mediante autorización, autenticación, protección, auditoría, entre otros.

1.10.2.4.1. APACHE KNOX

El Apache Knox Gateway es una puerta de enlace API REST para interactuar con los racimos de Hadoop. El Knox Gateway proporciona un único punto de acceso para todas las interacciones REST con racimos de Hadoop. En esta capacidad, Knox es capaz de proporcionar la funcionalidad valiosa de ayudar en el control, la integración, el seguimiento y la automatización de las necesidades administrativas y analíticas que son críticas en la empresa. [35]

- Autenticación (LDAP y proveedor de autenticación de Active Directory).
- Federación/SSO (encabezado HTTP basado federación de identidades).
- Autorización (Service Level Autorización).
- Revisión de cuentas.

Si bien hay una serie de beneficios para los clusters Hadoop sin garantía, Knox Gateway también complementa los kerberos garantizados cúmulo bastante bien. Junto con el aislamiento de la red adecuada de un cluster Hadoop Kerberos asegurado, Knox Gateway proporciona a la empresa con una solución que:

- Se integra bien con las soluciones de gestión de identidad de la empresa.
- Protege los detalles de la implementación de clúster Hadoop (hosts y puertos están ocultos a los usuarios finales).

1.10.2.4.2. APACHE RANGER

Ranger es un framework que permita, supervisar y gestionar la seguridad de datos completa a través de la plataforma Hadoop.

La visión con Ranger es proporcionar seguridad integral en todo el ecosistema Hadoop. Con la llegada de Apache Yarn, la plataforma Hadoop ahora puede soportar una verdadera arquitectura de conjunto de datos. La seguridad de los datos dentro de Hadoop necesita evolucionar para soportar múltiples casos de uso para el acceso de datos, mientras que también proporciona un marco para la administración central de las políticas y el seguimiento de acceso de los usuarios de seguridad. [36]

1.10.2.5. Operaciones

Son las herramientas que permiten monitorear, administrar y realizar operaciones sobre un clúster Hadoop.

1.10.2.5.1. APACHE AMBARI

Es un proyecto que facilita la gestión de Hadoop. Ofrece una interfaz web intuitiva y fácil de usar para la gestión de Hadoop y además proporciona una API REST.[37] Ambari permite a los administradores del sistema:

- Monitoriza el clúster Hadoop.
- Ofrece un panel de control para vigilancia de la salud y el estado del cluster Hadoop.
- Se encarga de la instalación de los paquetes de Hadoop en el clúster.
- Un asistente paso a paso para la instalación de servicios de Hadoop a través de múltiples equipos.
- Proporciona la forma de gestionar gestión central para iniciar, detener y volver a configurar los servicios de Hadoop en todo el clúster.

1.10.2.5.2. APACHE ZOOKEEPER

Zookeeper es un proyecto de Apache que proporciona una infraestructura centralizada y de servicios que permiten la sincronización del cluster. ZooKeeper mantiene objetos comunes que se necesitan en grandes entornos de cluster. [38]

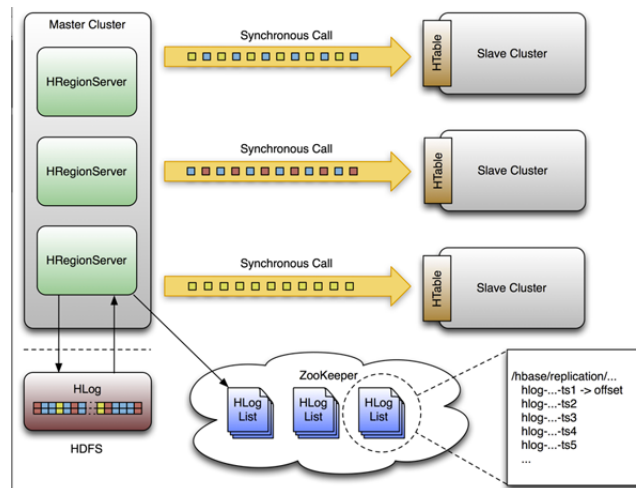


Figura 1.14: Zookeeper.

1.10.2.5.3. APACHE OOZIE

Apache Oozie es un sistema de programación de flujo de trabajo basado en servidor para administrar los trabajos de Hadoop. Los flujos de trabajo en Oozie

se definen como una colección de flujos de control y nodos de acción en un grafo dirigido acíclico.

Los nodos de flujo de control definen el comienzo y el final de un flujo de trabajo (inicio, final y los nodos de fallo), así como un mecanismo para controlar la ruta de ejecución del flujo de trabajo. Los nodos de acción son el mecanismo por el cual un flujo de trabajo provoca la ejecución de una tarea de cálculo/procesamiento.

Oozie se implementa como una aplicación web en Java que se ejecuta en un contenedor de servlets Java y se distribuye bajo la licencia Apache 2.0. [39]

1.10.3. Distribuciones HADOOP

En el siguiente capítulo se describirán algunas de las distribuciones Hadoop.

1.10.3.1. CLOUDERA

La distribución de Cloudera (CDH) fue la primera en aparecer en el mercado, combinando Big Data y Hadoop. CDH no solo incluye el núcleo de Hadoop (HDFS, MapReduce...) sino que también integra diversos proyectos de Apache (HBase, Mahout, Pig, Hive, etc.). CDH es open-source, y cuenta con una interfaz gráfica propietaria, Cloudera Manager, para la administración y gestión de los nodos del clúster Hadoop. La descarga es totalmente gratuita.

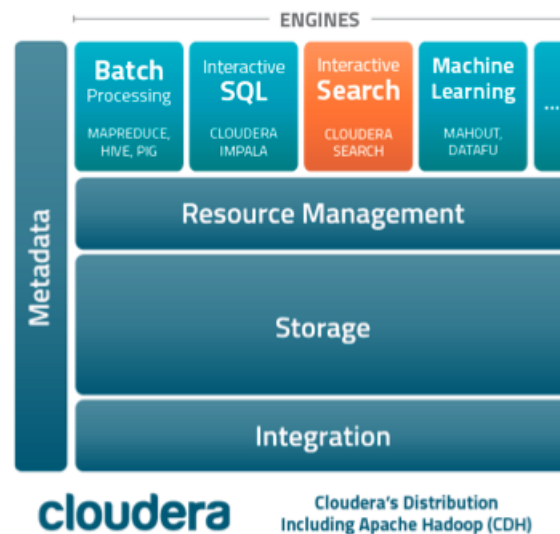


Figura 1.15: Cloudera.

No obstante, también cuenta con una versión empresarial, que incluye una

interfaz más sofisticada. Cloudera recientemente ha estrechado vínculos tanto con IBM como con Oracle. [40]

Cloudera ofrece software, servicios y soporte en 3 paquetes:

- Cloudera Enterprise incluye CDH (Cloudera Distribution Hadoop) y una licencia de subscripción anual (por nodo) a Cloudera Manager y soporte técnico. Hay 3 modos: Basic, Flex y Data Hub.
- Cloudera Express incluye CDH y una versión de Cloudera Manager sin características de negocio como rolling upgrades y backup/disaster recovery.
- CDH se puede descargar de la página de Cloudera, pero sin soporte técnico ni el Cloudera Manager.

1.10.3.2. MAPR

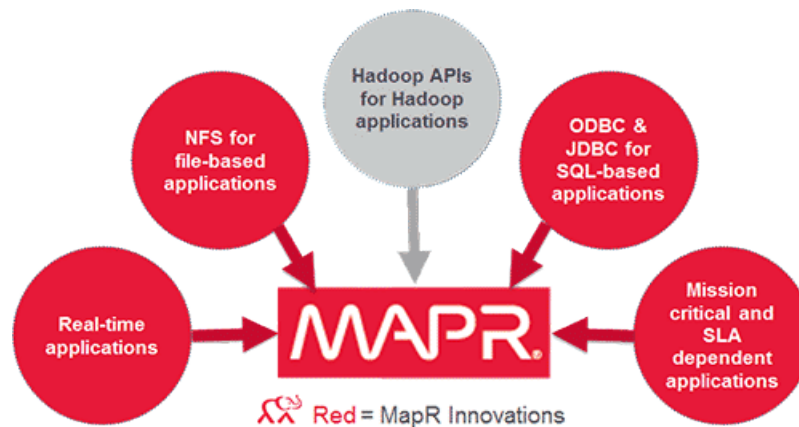


Figura 1.16: MapR.

Esta distribución posee una gama de soluciones con respecto al desarrollo sobre un ecosistema Hadoop. MapR ofrece soluciones para trabajar con bases de datos SQL sobre Hadoop, como también ofrece soluciones NoSQL. [41]

1.10.3.3. HORTONWORKS

Hortonworks es una de las distribuciones más recientes de Hadoop (HDP). Al igual que CDH, HDP es totalmente open-source, incluye las herramientas que forman el núcleo de Hadoop, y por supuesto también incorpora diferentes proyectos open-source de Apache. [42]

La plataforma de datos Hortonworks permite el despliegue de Open Enterprise Hadoop - el aprovechamiento de componentes de código abierto 100 %, la conducción de la empresa requisitos de preparación y se faculta a la adopción de nuevas innovaciones que sale de la Apache Software Foundation y proyectos clave de Apache.

Este amplio conjunto de capacidades se alinea con las siguientes áreas funcionales: gestión de datos, acceso a datos, el gobierno de datos y la integración, seguridad y operaciones.

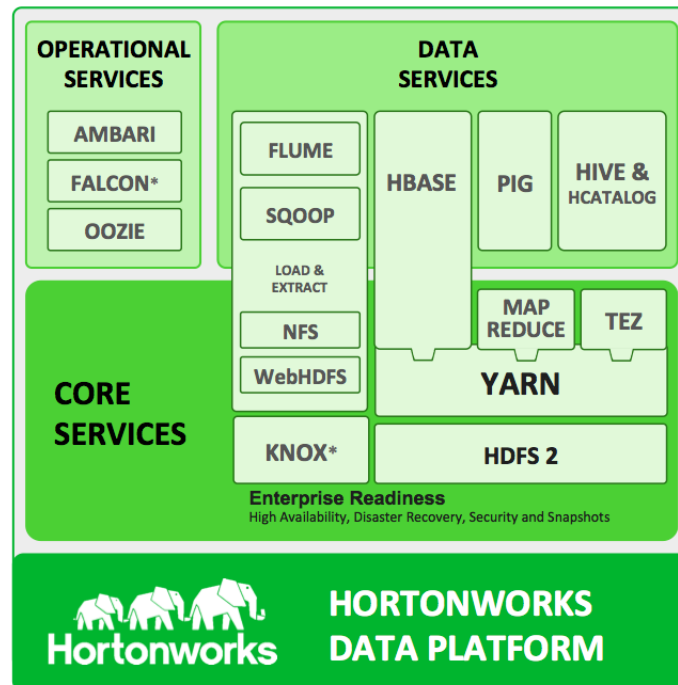


Figura 1.17: Hortonworks.

CAPÍTULO 2

REDES SOCIALES

2.1. Concepto de Red Social

Una Red Social es una estructura social integrada por personas, organizaciones o entidades que se encuentran conectadas entre sí por una o varios tipos de relaciones como ser: relaciones de amistad, parentesco, económicas, relaciones sexuales, intereses comunes, experimentación de las mismas creencias, entre otras posibilidades.[43]

Las redes sociales de internet se han convertido sin dudas en un fenómeno social que revoluciona la manera de comunicación y la interacción que hasta el momento teníamos los seres humanos. Algunas de las redes sociales más populares de la actualidad son: Facebook, Twitter, LinkedIn, Instagram, Google Plus+, entre otros.

2.2. Clasificación de las redes sociales

Las redes sociales han existido desde el comienzo de los tiempos. En cambio, la digitalización de éstas es muy reciente y en poco tiempo se han convertido en el fenómeno mediático de mayor envergadura. Para comprender la nueva realidad social debemos conocer en profundidad los diferentes tipos de redes sociales digitales en adelante, redes sociales que operan en la Red. Mencionaremos la siguiente clasificación:

2.2.1. Por su público objetivo y temática

- Redes sociales Horizontales: Son aquellas dirigidas a todo tipo de usuario y sin una temática definida. Se basan en una estructura de celdas permitiendo la entrada y participación libre y genérica sin un fin definido, distinto del de generar masa. Los ejemplos más representativos del sector son Facebook, Orkut, Twitter.

- Redes sociales Verticales: Están concebidas sobre la base de un eje temático agregado. Su objetivo es el de congregar en torno a una temática definida a un colectivo concreto. En función de su especialización, pueden clasificarse a su vez en:
 - Redes sociales Verticales Profesionales: Están dirigidas a generar relaciones profesionales entre los usuarios. Los ejemplos más representativos son Viadeo, Xing y LinkedIn.
 - Redes sociales Verticales De Ocio: Su objetivo es congregar a colectivos que desarrollan actividades de ocio, deporte, usuarios de videojuegos, fans, etc. Los ejemplos más representativos son Wipley, Minube Dogster, Last.FM y Moterus.
 - Redes sociales Verticales Mixtas: Ofrecen a usuarios y empresas un entorno específico para desarrollar actividades tanto profesionales como personales en torno a sus perfiles: Yuglo, Unience, PideCita.

2.2.2. Por el sujeto principal de la relación

- Redes sociales Humanas: Son aquellas que centran su atención en fomentar las relaciones entre personas uniendo individuos según su perfil social y en función de sus gustos, aficiones, lugares de trabajo, viajes y actividades. Ejemplos de este tipo de redes los encontramos en Koornk, Dopplr, Youare y Tuenti.
- Redes sociales de Contenidos: Las relaciones se desarrollan uniendo perfiles a través de contenido publicado, los objetos que posee el usuario o los archivos que se encuentran en su ordenador. Los ejemplos más significativos son Scribd, Flickr, Bebo, Friendster, Dipity, StumbleUpon y FileRide.
- Redes sociales de Inertes: Conforman un sector novedoso entre las redes sociales. Su objeto es unir marcas, automóviles y lugares. Entre estas redes sociales destacan las de difuntos, siendo éstos los sujetos principales de la red. El ejemplo más llamativo es Respectance.

2.2.3. Por su localización geográfica

- Redes sociales Sedentarias: Este tipo de red social muta en función de las relaciones entre personas, los contenidos compartidos o los eventos creados. Ejemplos de este tipo de redes son: Rejaw, Blogger, Kwippy, Plaxo, Bitacoras.com.
- Redes sociales Nómadas: A las características propias de las redes sociales sedentarias se le suma un nuevo factor de mutación o desarrollo basado en la localización geográfica del sujeto. Este tipo de redes se componen y recomponen a tenor de los sujetos que se hallen geográficamente cerca del lugar en el que se encuentra el usuario, los lugares que haya visitado

o aquellos a los que tenga previsto acudir. Los ejemplos más destacados son: Latitud, Brighkite, Fire Eagle y Skout.

2.2.4. Por su plataforma

- Red Social Metaversos: Normalmente contruidos sobre una base técnica Cliente-Servidor WOW, SecondLife, Lineage, Gladius, Travian y Habbo.
- Red Social Web: Su plataforma de desarrollo está basada en una estructura típica de web. Algunos ejemplos representativos son: MySpace, Friendfeed y Hi5.

2.3. Análisis de Redes Sociales

Las empresas cada vez se preocupan más de la obtención y recolección de datos, lo que genera la necesidad de tener sistemas que capturen, almacenen, busquen, compartan, analicen y visualicen grandes conjuntos de datos (Big Data), con el fin de procesar todos los datos disponibles, sacar información de los datos, tomar decisiones con la información. Para desarrollar una solución Big Data en el área de las redes sociales, es importante:

- Definir claramente que vamos a solucionar usando Big Data.
- Definir la mejora que se quiere obtener y como utilizar la información.
- Conocer “que conoces” y “que no conoces”.
- Conocer a los clientes, conocer la competencia.
- Conocer lo que realmente piensan los clientes.
- Escoger la herramienta adecuada.
- Saber que se quiere analizar, ayudará a escoger la herramienta.
- Testea las hipótesis.
- Una vez tengamos datos, analizar desde diferentes perspectivas.
- Crear las ideas y accionarlas.
- De los datos recibidos, las hipótesis testeadas, deben salir nuevas percepciones que lleven a realizar acciones para conseguir solucionar las necesidades iniciales.

2.4. Arquitectura Big Data para Redes Sociales

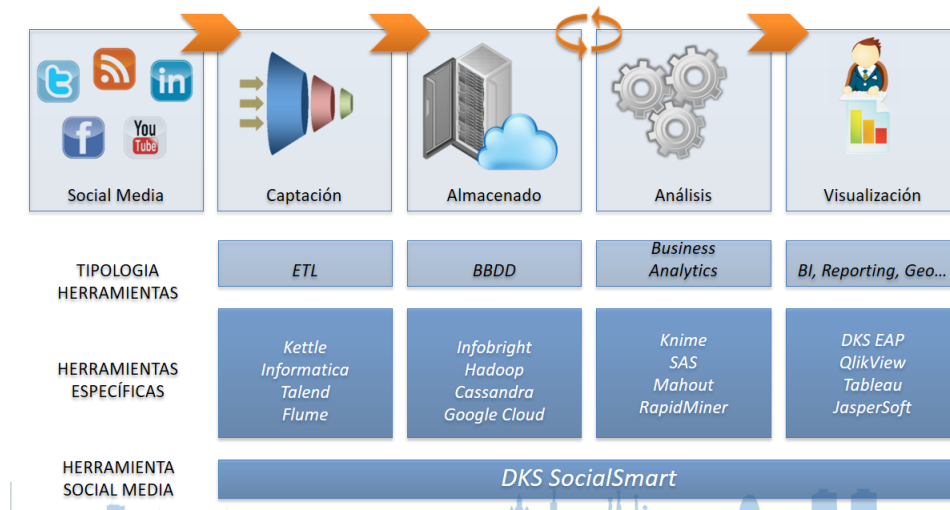


Figura 2.1: Ejemplo de una arquitectura Big Data para redes sociales.

2.4.1. Captación

- Orígenes de Datos: Los orígenes de datos pueden ser cualquier fuente de una red social u otra: Redes Sociales, foros, blogs, analytics, mailings, ventas, entre otros.
- Captación: Cada origen de datos puede requerir una manera de captación de datos, por ejemplo: Twitter: API pública, GNIP, DataSift, etc. Facebook: GraphAPI, etc. Blogs: Herramientas de Scraping como 80legs, rss, etc.
- Características a tener en cuenta:
 - Volumen: Cantidad de “mensajes” que podemos recoger.
 - Velocidad: Tiempo de recolección de los mismos.
 - Antigüedad: Acceso a datos pasados.
 - Privacidad: Acceso a datos privados.
 - Precio: Que cuesta cada “mensaje”.
 - Fiabilidad: Que cantidad de “mensajes” se recoge respecto al total.

2.4.2. Almacenado

Debido a la incapacidad de las bases de datos distribuidas (BDD) tradicionales de dar solución a los retos de los grandes volúmenes de datos, aparecen

diferentes tipologías de BDD para dar solución a sistemas “Big Data”. Hay una tipología de BDD idónea para cada tipo de negocio:

Algunas Tipologías de BDD “Big Data”:

- BDD Analíticas: BBDD idóneas para optimizar el tiempo de Consulta SQL. Implantación real: Yahoo la utiliza para calcular el precio de sus anuncios.(Hadoop)
- BDD Documentales: BBDD idóneas para la gestión de “documentos”. Implantación real: Foursquare utiliza para check-in e información geo (MongoDB).
- BDD en Grafo: BBDD idóneas para datos con relaciones indeterminadas. Implantación real: Google la utiliza para relacionar páginas web. (Plegel)
- BDD Clave/Valor: BBDD idóneas para aplicaciones on-line. Implantación real: Twitter la utiliza para su aplicación web. (Cassandra)

2.4.3. Análisis

En esta fase se aplica el análisis de texto o Text Analytics, que es un proceso que permite un análisis automático y unificado de textos provenientes de diferentes fuentes de datos como plataformas de redes sociales, blogs o páginas de noticias. Permite la detección del idioma, sentimientos, identificación de cadenas léxicas, categorización de textos en diferentes temas y descubrimiento de trending topics. Ejecuta los siguientes pasos:

- Limpia el texto de enlaces, jergas, entre otros.
- Corrige ortográficamente las palabras.
- Asigna categorías gramaticales.
- Reconoce personas, lugares, organizaciones.
- Relaciona e identifica dependencias.

2.4.4. Visualización

La información correcta debe llegar a la persona correcta, en el momento correcto, en el formato correcto, en el dispositivo correcto, con el detalle correcto, etc. Es decir la distribución y presentación de la información. Hay muchas maneras de representar la información:

- Cuadros de Mando: QlikView, Tableau.
- Informes: JasperReports, CristalReports.
- Gráficos tradicionales: Any Charts, Google Charts.

- Mapas: DKS GeoSmart, ESRI.
- Gráficos a medida: Flash.
- Infografías.

2.5. Módulos del sistema Big Data para Redes Sociales

Pueden existir diversos módulos para un sistema Big Data en el área de las Redes Sociales, en este documento trataremos 2 módulos.

2.5.1. Módulo I: Análisis de Redes Sociales

Es el módulo en el cual utilizando la teoría de grafos y datos generados en las redes sociales permitirán la extracción de conocimiento (minería de datos). El análisis de redes sociales se fundamenta en:

- Detectar de comunidades y grupos.
- Estudiar los agentes en la estructura de la red con el objetivo de ver las relaciones de poder, confianza, protagonismo, entre otros.

2.5.1.1. Objetivo principal del Análisis de Redes Sociales

Obtener información mediante las principales redes sociales para la realización de diversas mediciones estructurales y dinámicas. A partir de la información generada por este módulo, se obtendrán los insumos necesarios para la realización de actividades de monitoreo y planificación estratégica sobre las redes sociales.

2.5.1.2. Principales Funciones para el Análisis de Redes Sociales

2.5.1.2.1. Recopilación de datos Proceso de obtener datos acerca de los usuarios de las redes sociales y la información que estos publican para posteriormente procesarla y analizarla.

2.5.1.2.2. Identificación de comunidades

Proceso que permite identificar los grupos en las redes sociales con la finalidad de calcular el tiempo de difusión o propagación de una información y pronosticar el volumen e impacto de una noticia. Los métodos de agrupamientos permiten clasificar:

- Comunidades y usuarios.
- Comunidades influyentes.
- Post/ Tuits (Traducción de la palabra en inglés Tweets).

2.5.1.2.3. Identificación de usuarios influyentes

Mediante las mediciones de centro es posible identificar los actores más influyentes en las redes sociales. De su conocimiento y ubicación dentro de las diferentes agrupaciones, se pueden trazar estrategias para la gestión de diversas situaciones. Cualquiera que emite o comunica en las redes sociales es un influencer o influyente desde el momento en que puede llegar a ejercer influencia en su audiencia. Existen varios tipos de usuarios influyentes:

- Falso influenciador, robots y otros artilugios estratégicos: Detectar a este tipo de usuario no siempre es sencillo. En ocasiones utilizan perfiles con nombres no personales, pero en otros casos incluso se ocultan tras fotografías de personas reales. La proporción entre número de seguidores y seguidos puede ser un indicador, aunque no es suficiente.

Lo más efectivo es analizar la red en su conjunto y detectar patrones de comportamiento. Por ejemplo:

- Retuits coordinados de un tema, esto es, retuits que realiza de manera casi conjunta un grupo de usuarios con unas diferencias de tiempo que parecen programadas.
- Emisión de información a una frecuencia superior a la de una persona real, aunque esto también es complicado, pues hay usuarios que parece que respiran y tuitean a la vez (hiperactivos digitales).

Un ejemplo de red social de robots mostraría una estructura de enlaces entre nodos, es decir, muchos nodos interactuando sobre un tema común pero sin relaciones entre ellos, algo antinatural, pues cuando los seres humanos tienen intereses comunes tienden a organizarse en grupos de interés, lo que en las redes sociales se mostraría como estructuras de nodos unidas en mayor medida por lazos de seguido/seguidor.

- Influenciador “autobombo”: Son usuarios que vienen a hablar de algo que les pertenece o conviene, y no hablan de otra cosa que no sea de eso. En el caso de que sea necesario detectarlos y se trata de organizaciones, es más fácil detectarlas, pues suelen seguir un horario y fuera de él no realizan actividad digital alguna, por lo que cuando analizas la conversación que gira en torno a su marca, puedes detectar claras variaciones a partir de determinadas horas de la tarde o los fines de semana.
- Influenciador-generador de opinión: Son los usuarios que crean la información original, o el rumor. Se les llama también “iniciadores de ideas” (idea starters en inglés), pues son ellos los que ponen en circulación las ideas de las que los demás se harán eco. Son uno de los tipos principales de influencers, pues sus ideas, o rumores, pueden generar corrientes nuevas de pensamiento, lo que conocemos como trending topics.

Cualquiera puede ser un generador de opinión, no es necesario que tenga muchos seguidores, ni es necesario que haya sido un líder de opinión anteriormente, sólo tienen que confluir dos circunstancias:

- Tener una idea original.
 - Viralizar la idea.
- **Influenciador amplificador:** Usuarios amplificadores son aquellos que comparten información de otros y hacen que llegue a mucha más gente. Generalmente tienen muchos seguidores y además seguidores que se hacen mucho eco de lo que dicen. Éste es el tipo que comunmente se conoce como influencer, pues es el que más alto índice tiene en diferentes indicadores como número de seguidores, Klout, page rank, etc. Su principal función en la red es la de amplificar la información creada por los generadores de opinión y hacer que llegue a un mayor número de personas. Es decir, son los que otorgan influencia a los generadores de opinión. Se tienen de varias clases, al igual que en los generadores de opinión. Sin ser exhaustivo, tenemos los siguientes:
- El radio patio por excelencia: todo lo que le llega lo retrasmite. No aporta información, no se posiciona respecto a la información y seguramente ni se lee la información. Este tipo de usuario hace de distribuidor masivo de información, si bien por su idiosincrasia, aunque potencialmente llegue a mucha gente, posiblemente no le hagan demasiado caso.
 - El experto en un tema que retrasmite todo lo que le llega de ese tema siempre que le interese. En ocasiones hace aportes personales y se puede posicionar con respecto a esa información. Aunque el número de contactos de este tipo fuera menor, su credibilidad hace que el nivel de interacción que tienen sus contactos con él sea mayor, por lo tanto suele tener mayor alcance que el primero. Este tipo de usuario es el que nos interesa, pues es el que va a dar una visión real de nuestro producto y va alcanzar a una audiencia que confía en él.
 - El que utiliza su posición en la red para su propio beneficio. Por ejemplo, incrementar su visibilidad y posicionamiento como gurú de un tema. Este usuario no hará nada que no sea favorable a su posicionamiento y se posicionará según lo que considere que es políticamente correcto y no su propia opinión. Es el tipo que me gusta llamar “falso gurú”. Este tipo de usuario es el que más influencia suele tener, pues por estrategia ha sabido posicionarse bien.
- **Influenciador distribuidor:** Las redes sociales tienden a organizarse mediante estructuras fuertemente cerradas, esto es, generalmente la red se organiza en módulos o grupos de usuarios más o menos interconectados entre sí pero que no tienen demasiada conexión con otros grupos. Esto es lo que se conoce por organización basada en comunidades. Los usuarios distribuidores son aquellos que interconectan dos o más comunidades y por lo tanto sirven de puente entre dos agrupaciones de usuarios que de otro modo permanecerían interconectadas. Estos usuarios son los que permi-

ten que la información trascienda, salga de su ámbito y pueda convertirse realmente en un trending topic.

Este tipo de usuarios es el que tienes que tener bajo el punto de mira en caso de que se produzca una crisis, pues son los que pueden hacer que esa crisis se viralice y se salga de tu ámbito o dominio. Si se produce una crisis los usuarios implicados van a buscar a estos usuarios para tratar de difundir su mensaje, por ejemplo, poniéndolos en copia en sus tuits para que se hagan eco y así trascender las fronteras de su comunidad.

2.5.1.2.4. Análisis de tendencias

La evaluación permanente de la evolución de los mensajes que fluyen a través de las redes sociales es posible medir tendencias y estar en conocimiento sobre los que se "dice" la opinión que tienen la gente sobre diversos tópicos.

2.5.1.3. Métricas

Es importante definir métricas cuantitativas como métricas cualitativas:

- Métricas cuantitativas nos aportarán una idea de la extensión y el impacto que tienen las redes sociales.
- Métricas cualitativas son las que realmente le tomarán el pulso al rendimiento que las redes sociales aportan y el interés que generan en los usuarios.

Ejemplos:

- Principales métricas en Facebook:
 - Número de Likes a la página y su crecimiento.
 - Número, tipo y sentimiento de los comentarios de los visitantes.
 - Visitas a la página.
 - Porcentaje de comentarios respondidos y tiempo de respuesta.
- Principales indicadores en Twitter:
 - Número y crecimiento de seguidores.
 - Número de retweet y menciones.
 - Sentimiento de las menciones y tiempo de respuesta.
 - Número de seguidos.
 - Número de tweets y periodicidad de envío.
 - Horas en las que se producen mas interacciones.

2.5.1.3.1. Principales métricas para el Análisis de Redes Sociales

- Volumen: Mide la cantidad de mensajes sobre un determinado tema, así como la cantidad de personas que hablan de dicho tema. También se debe dar seguimiento a cómo cambian estas cifras con el paso del tiempo.
- Alcance: Mide la difusión de la conversación en redes sociales, además del tamaño potencial de la audiencia que ayuda a contextualizar otras métricas de compromiso.
- Compromiso: En la mayoría de las redes sociales, el contenido se puede compartir, así como responder (ej. Retweets (RT)), los mismos que sirven para saber quién divulga su contenido.
- Influencia: Mide el capital social en línea y la capacidad para influir en otros. Cabe destacar que el tamaño de la audiencia no necesariamente se relaciona con la influencia. Solo porque alguien tiene muchos seguidores no significa que puedan alentarlos para que de verdad hagan algo.
- Participación de voz (Share of voice) Nos permite determinar cuánto porcentaje de la conversación total está centrado sobre un tema específico, en comparación con otros temas.
- Sentimiento: Este análisis consiste en saber cuál es el sentimiento de los usuarios sobre un determinado tema. En el análisis de sentimiento de la información se debe enfocar en determinar si los usuarios de las redes sociales (Twitter) están de acuerdo, desacuerdo o les es Indiferente el tema.

2.5.2. Módulo II: Monitoreo de Redes Sociales

Módulo que permitirá conocer:

- Menciones, opiniones, interacciones, alcance, popularidad e impacto de nuestra marca, producto, servicio y/o programa.
- Temas de interés en un momento dado.
- Tendencias generadas Internet por los distintos usuarios.

2.5.2.1. Objetivo Principal del Monitoreo de Redes Sociales

El objetivo es no perder el control del principal medio de expresión de las personas. El 53 % de los usuarios prefiere utilizar las redes sociales para realizar consultas o quejas a una empresa o marca. El 65 % de los usuarios aprende sobre una empresa o marca gracias a las redes sociales. En este contexto es importante poder establecer un plan de acción para poder escuchar y actuar ante lo que clientes y/o usuarios conversan y opinan sobre una marca, empresa o temática en las diferentes redes sociales. Monitorear mediante mecanismos de escucha activa el estado de las redes sociales más utilizadas en el país (Facebook, Twitter, Instagram y LinkedIn), en cuanto a:

- Los temas más tratados.
- La identificación de fuentes de información que fluyen por la red.
- La identificación de amenazas potenciales.

Todo esto con la finalidad de facilitar la supervisión de tendencias de opinión pública y de situaciones de crisis.

2.5.2.2. Principales Funciones para el Monitoreo de Redes Sociales

2.5.2.2.1. Monitoreo Inteligente Permite hacer seguimiento de las conversaciones importantes y las tendencias en varias redes sociales. Es una herramienta útil en la gestión de crisis que emitirá reportes en relación a:

- Palabras claves.
- Frases.
- Etiquetas (“Hashtags”).
- Cuentas de usuarios importantes que deben ser permanentemente monitoreados.

2.5.2.2.2. Medición de usuarios nacionales

Permite detectar quiénes son los usuarios de Twitter del país. Se hace utilizando información libre y abierta que permite inferir con un alto grado de acierto el país de procedencia.

2.5.2.2.3. Medios locales y mundiales

Monitoreo todos los medios nacionales que tienen presencia online. Se basa un lector específico para cada medio nacional. Permite identificar dónde se han publicado noticias relevantes sobre un tema de interés.

2.5.2.2.4. Selección de temas importantes

Identificación todas las conversaciones (Twitter), profundizando en los temas o tendencias que más interesan. La solución se basa en herramientas de agrupamiento (“clustering”) que ayudan a identificar cuáles son los temas más relevantes en la en cada momento.

2.5.2.2.5. Alertas y reportes automáticos

Aplicación para diseñar y enviar automáticamente mensajes de alerta para la organización. Las alertas se diseñan bajo criterios específicos de la organización y se envían en el mismo instante en que la información ingresa a la base de datos y es procesada.

2.6. Indicadores Big Data para Redes Sociales

En el Social Media hay indicadores clave como ROI, engagement, brand value, reputación, influenciadores, entre otros.

2.6.1. Retorno de la inversión en Social Media (ROI)

Es la relación de los INDICADORES de Social Media con el MARKETING más tradicional.

Leads = ALCANCE.

Impresiones = ALCANCE x ACTIVIDAD.

Interacción = AMPLIFICACIÓN, ADHESIÓN, CONVERSACIÓN.

Patrocinio = MENCIONES de terceros.

- Leads: CPL (Coste por Lead): coste de adquirir un contacto.
- Impresiones: CPM (Coste por mil): coste de producir mil impactos en el público objetivo.
- Interacción: CPI (Coste de Interacción) Valor de las interacciones con los consumidores.
- Patrocinio. El coste es variable en función de la relevancia de la institución o de la persona.

2.6.2. Compromiso (ENGAGEMENT)

Es la interacción directa de la comunidad con nuestra estrategia. Que porcentaje de usuarios realmente interactúan:

$$\frac{ADHESION + CONVERSACION + COMENTARIO \times 100}{ALCANCE} \quad (2.1)$$

2.6.3. Valor de la marca (BRAND ASSET VALUATION)

Es el análisis que mide el valor de cada COMPETIDOR respecto a los demás.

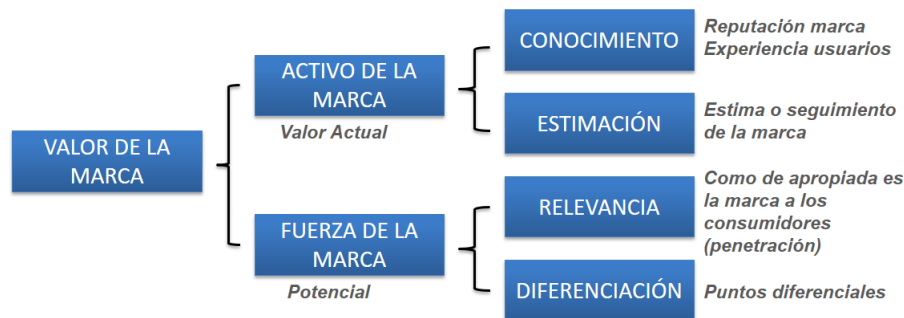


Figura 2.2: Valor de la marca.

2.6.4. Reacciones de consumo (CONSUMER REACTIONS)

Es el análisis de la actividad y MENCIONES realizadas por los USUARIOS. Analiza los mensajes realizados por USUARIOS donde se comenta:

- DIRECTAMENTE un PERFIL
- INDIRECTAMENTE una PALABRA CLAVE

La MENCIÓN es un MENSAJE donde su AUTOR es cualquier USUARIO de la PLATAFORMA, y en el MENSAJE se menciona o cita directamente un PERFIL del COMPETIDOR o una PALABRA CLAVE del COMPETIDOR.

- MENCIÓN DIRECTA
Mención donde se cita explícitamente el perfil o se escribe en su muro para dejar claro que se quiere referirse a el perfil y que este reciba el mensaje.

DIRECTA: Cuando mencionan un PERFIL.

DIRECTA: Cuando escriben en el muro del PERFIL.

- MENCIÓN INDIRECTA
Mención donde se cita explícitamente una palabra clave que es seguida por un competidor. El autor del mensaje no busca referirse directamente al perfil y que este reciba el mensaje.

INDIRECTA: Cuando cualquier usuario menciona una PALABRA CLAVE en su red social.

INDIRECTA: Cuando cualquier usuario menciona una PALABRA CLAVE en su POST.

En función del contenido del MENSAJE, una MENCIÓN tiene un SENTIMIENTO que puede ser POSITIVO, NEUTRO o NEGATIVO.

- POSITIVO: Cuando la MENCIÓN habla positivamente del PERFIL que menciona.
- NEUTRO: Cuando la MENCIÓN simplemente comenta sin querer influenciar de manera positiva o negativa sobre el PERFIL que menciona.
- NEGATIVO: Cuando la MENCIÓN habla negativamente del PERFIL que menciona.

2.6.5. Seguidores (BRAND AUDIENCE)

Es el análisis de los SEGUIDORES de los PERFILES de un COMPETIDOR. Se centra en analizar diferentes aspectos de los seguidores de los perfiles de un competidor:

- PERFIL DE LOS SEGUIDORES DE UN COMPETIDOR.
- COMPARATIVA DE SEGUIDORES ENTRE COMPETIDORES.
- MENSAJES DE LOS SEGUIDORES.

SEGUIDORES ÚNICOS

De todos los SEGUIDORES de los PERFILES de un COMPETIDOR se cuentan sin repetidos.

ESPECIFICIDAD DEL PERFIL

Número de SEGUIDORES que solo siguen el PERFIL de un COMPETIDOR.

2.6.6. Influyentes (INFLUENCERS)

El análisis de los AUTORES que realizan MENCIONES DIRECTAS o INDIRECTAS de un COMPETIDOR. Se centra en analizar diferentes aspectos de los AUTORES que realizan MENCIONES de un COMPETIDOR.

- INFLUENCIADORES POR ALCANCE O ACTIVIDAD.
- ANALISIS DE RELEVANCIA.
- ANALISIS DE SENTIMIENTO.
- INDICE KLOUT.
- EMBAJADORES, TROLLS.

2.6.7. Contenidos

El análisis de todo el CONTENIDO de los MENSAJES que tienen MENCIONES INDIRECTAS (que se ha hecho referencia a una PALABRA CLAVE de un COMPETIDOR en el MENSAJE). Se centra en analizar los CONTENIDOS de los MENSAJES con MENCIONES INDIRECTAS para detectar las palabras que se utilizan en ellos, a lo que llamaremos TOPICS.

- **TOPIC** Palabras simples o compuestas que aparecen en el CONTENIDO de los MENSAJES que tienen MENCIONES INDIRECTAS de PALABRAS CLAVE que siguen los COMPETIDORES.
- **TRENDING TOPIC** TOPIC más repetidos según temporalidad y localización.
- **CATEGORÍAS** Las CATEGORÍAS sirven para agrupar TOPICS.

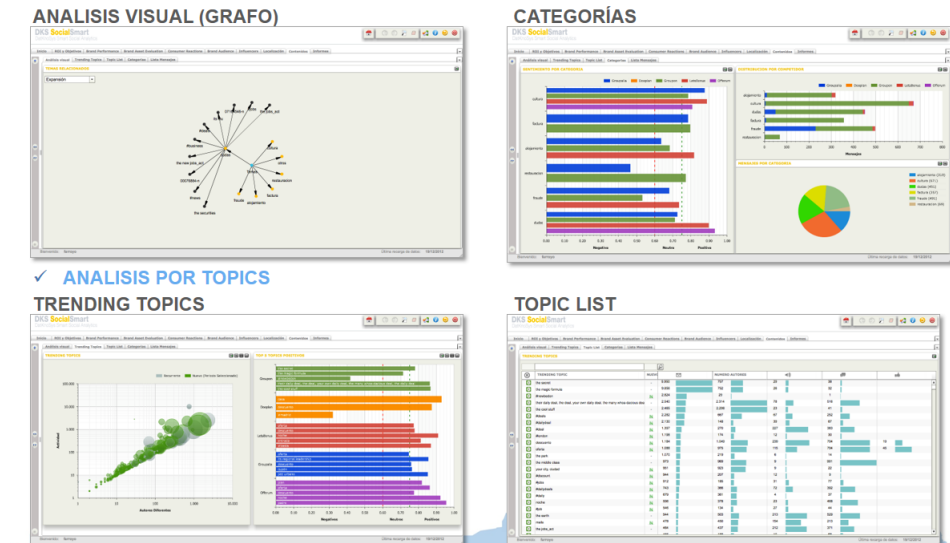


Figura 2.3: Análisis por categorías

2.6.8. Localización

Es el análisis de la ubicación de los MENSAJES LOCALIZADOS.

- **MENSAJE LOCALIZADO**: MENSAJE que ha compartido su ubicación en el momento de la publicación.
- **PAÍS**: PAÍS desde donde se publica un MENSAJE.
- **POBLACIÓN**: POBLACIÓN desde donde se publica un MENSAJE.
- **IDIOMA**: IDIOMA en que está escrito un MENSAJE.

2.7. Redes sociales en Venezuela

De acuerdo con el Monitor País de Hinterlaces sobre Penetración de Medios de Comunicación y Redes Sociales en el 2015, un 27 % de los venezolanos tiene una cuenta en Twitter, frente a 70 % que no la posee. Un 3 % de los 1.200 entrevistados no sabe o no contesta con respecto a este punto. El 27 % de los usuarios de Twitter accede diariamente a esa red, 25 % de ellos sigue al menos a una marca, 52 % twitea al menos una vez al día, y 37 % accede desde un teléfono al sistema de mensajes de 140 caracteres.[44]

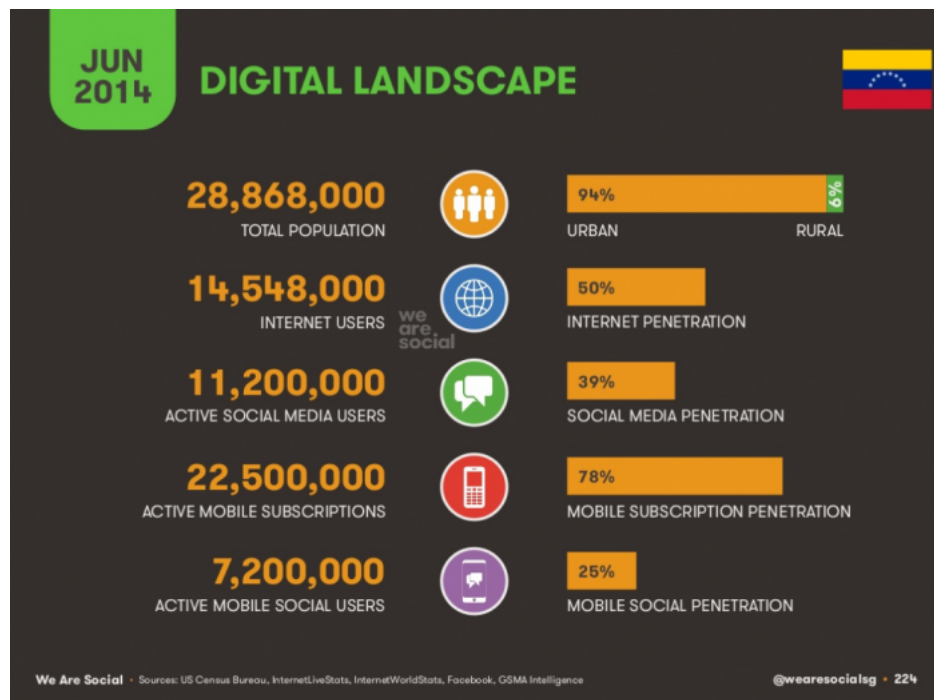


Figura 2.4: Estadísticas de redes sociales en Venezuela.

Si se considera que 69 % de los venezolanos cuenta con servicio de suscripción a Internet en su hogar, la cifra correspondiente a quienes mantienen una cuenta en Twitter es menor a la mitad de los suscriptores a la web. Sin embargo, no todos los que mantienen una cuenta en esta red social la utilizan en forma regular, participan, opinan o interactúan. El grado de influencia de los usuarios también es un factor adicional que no está reflejado en el estudio que se menciona, pero que tiende a activarse de acuerdo con banderas políticas de lado y lado, en especial cuando se trata de redes organizadas que buscan impulsar etiquetas o hashtags ante eventos específicos con la finalidad de impactar en la opinión pública local e internacional.

Con respecto a Facebook, 9.7 millones de venezolanos están conectados a Facebook, lo que se convierte en un 33 % de penetración de esta red social en la población. De estos 9.7 millones de venezolanos, 41 % accede a diario; 40 % sigue a una marca, sólo 12 % actualiza su estatus diariamente y 30 % interactúa usando un celular.

LinkedIn es menos popular (100 millones totales, 500 mil en Venezuela) pero tiene un foco preciso hacia la divulgación e intercambio de perfiles profesionales. .^{E1} "amigo de tu amigo en LinkedIn es tu próximo jefe" dice un popular lema sobre esta red.

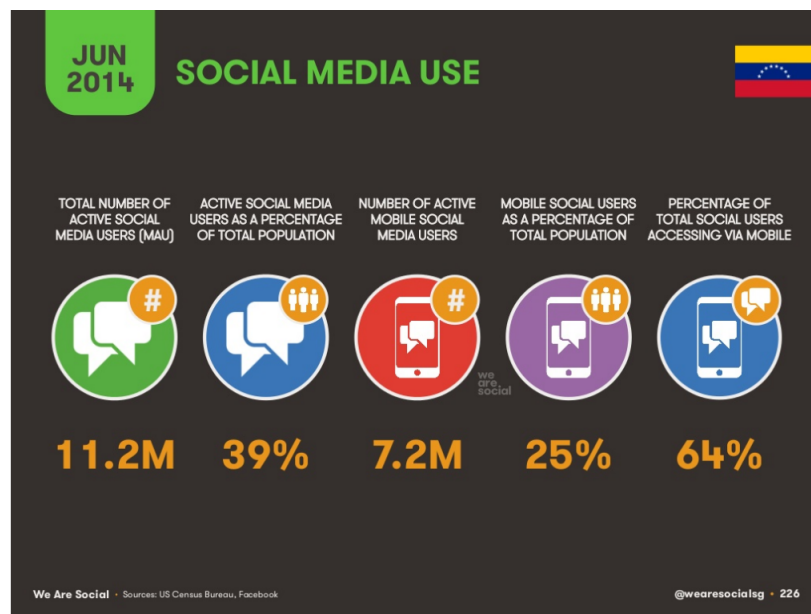


Figura 2.5: Estadísticas de redes sociales en Venezuela.

Más de 7 millones se conectaron a las Redes Sociales desde su teléfono o tableta en el 2014. Los usuarios de internet en Venezuela eran 14.548.000 eso representa el 50 % de penetración en la población. 11.200.000 fueron usuarios activos de Redes Sociales, para un 39 % de penetración. 7.200.000 fueron usuarios móviles sociales activos, lo que significa un 25 % de penetración social y móvil. Un 64 % de los usuarios activos en redes sociales accede vía móvil. 93 % del total de los suscriptores móviles son pre-pago, 7 % son post-pago. 36 % se conecta vía 3G. [45]

CAPÍTULO 3

Análisis de redes sociales

3.1. Análisis de redes sociales o social network analysis (SNA)

Es un área de investigación que estudia las redes sociales como grafos, en un intento por hacer sociología de forma precisa y explicar la macrosociología a partir de la microsociología. Las redes sociales están conformadas por un conjunto de usuarios relacionados entre sí, por ende podemos definir un grafo donde los vértices son los usuarios y las aristas los vínculos.

Con esto hemos definido un grafo, pues tenemos los vértices y las aristas o, como le llaman los sociólogos, actores y vínculos. Los actores pueden ser personas o grupos de éstas: empresas, comunidades, organizaciones de apoyo social, países, ciudades, etc. Los vínculos son cualquier cosa que relacione a los actores, por ejemplo: amor, poder, alianzas, amistad, parentesco familiar, contacto por correo electrónico, creencias religiosas comunes, rivalidad, etc.

3.2. Pasos para realizar un análisis de redes sociales

En primer lugar es necesario obtener la fuente de datos y representarla como un grafo. Luego, analizamos el grafo para determinar propiedades de la red social original. Existen diversas formas de analizar esta información.

3.2.1. Estudiando las características generales

Es posible que existan muchos tipos de redes sociales que faltan por ser clasificadas, pero se puede observar características comunes en ellas.

3.2.1.1. Redes de mundo pequeño (small world networks)

Ocurre en las redes que tienen una conectividad especial que hace que la distancia promedio entre dos actores cualquiera sea muy pequeña en comparación con el tamaño (número de actores) de la red. Un ejemplo de esto es un estudio que realizó Stanley Milgram, un importante psicólogo norteamericano, donde medía la distancia promedio entre personas, en redes de contacto. Eligió personas de Omaha (Nebraska) y Wichita (Kansas) para que se contactaran con personas de Boston (Massachusetts). La gente de Omaha y Wichita indicaba si conocían, o no, a las personas de Boston. En caso contrario, remitían a un contacto que pudiera conocerlas, con las que se repetía el proceso. Milgram, informado de todo esto, pudo medir cual era el largo de los caminos recorridos. El resultado: 6 personas de distancia en promedio.

3.2.1.2. Redes libres de escala (scale free networks)

Tienen una distribución de grados que sigue una ley de potencias o similar porque si tomamos un subgrafo de esta red, lo más probable es que los grados se sigan distribuyendo como ley de potencias. Las redes libres de escala son interesantes porque son regidas por leyes de potencia, que se repiten en otros casos como en la distribución del ingreso.

3.2.2. Estudiando la posición de los actores

El concepto de localidad de un actor en una red corresponde al acceso que tiene al resto de la red. En principio, sabemos que dos actores ocupan el mismo lugar en la red si comparten los mismo vecinos (equivalencia estructural, una versión local de isomorfismo de vértices). Pero en general deseamos ir más lejos. En esta necesidad definimos las medidas de centralidad, que miden la posición de un actor en una red de acuerdo a ciertos criterios.

3.2.2.1. Centralidad de grado (degree centrality)

En términos de grafos, la centralidad de grado de un actor se calcula como su número de vecinos. Si estamos modelando una red social de amigos, la centralidad de cada actor consiste en su número de amigos. Por ejemplo, Un hombre o mujer popular es aquel que tiene muchos amigos o conocidos.

3.2.2.2. Centralidad de Bonacich

Hay gente cuya favorable posición en una red social les permite iniciar procesos influenciales como la transmisión de creencias, chismes (gossip), publicidad viral, etc. En estos casos, el proceso empieza en un actor y se distribuye a su vecinos, los cuales redistribuyen a sus propios vecinos, sucesivamente. Entonces, podemos proponer una medida de centralidad para tales situaciones, que consista en contar los caminos. Sin embargo, las redes con ciclos nos dan problemas pues tienen infinitos caminos. Por eso, no podemos contar los caminos así nada

más. La solución práctica a este dilema es atenuar los caminos usando una tasa de descuento, tal como se usa en la evaluación económica, las series de potencias, etc.

Así, los caminos más largos se suman como números más pequeños, y los infinitos son cero. Usar una tasa de descuento que hace más pequeños los caminos más grandes tiene ventajas conceptuales. Los procesos influenciales como los chismes, las creencias, etc. pueden ser muy efectivos en distancias cortas, pero su difusión se hace menos efectiva (o lenta) a grandes distancias. Ajustando la tasa de descuento, se puede simular cuan rápido se atenúa un proceso de difusión.

3.2.2.3. Centralidad de vector propio (eigenvector centrality)

Digamos que un actor central es aquel que tiene un vecindario con buena centralidad. Por ejemplo, un feriante puede conocer a mucha gente común mientras que un político puede conocer menos gente, pero que son personas influyentes. Al final, el político está mejor posicionado en influencia que el feriante, aunque conozca menos gente. Ahora, no todo el vecindario de un actor contará con la misma centralidad, por lo que hay que considerar que los actores con mayor centralidad son más influyentes que el resto. Siguiendo el concepto de centralidad recursiva, diremos que la centralidad de un actor es proporcional a la suma de las centralidades de sus vecinos en el grafo

3.2.2.4. Centralidad de cercanía (closeness centrality)

Otra medida de posición sale de considerar la distancia promedio al resto de la red. El actor que está más cerca de todo otro elemento de la red es el más central.

3.2.2.5. Centralidad de intermediación (betweenness centrality)

La centralidad de intermediación es el número de rutas mínimas en las que el actor participa. Por ejemplo, en estrategias militares y terrorismo, es importante distinguir los actores claves. Si son atacados, desconectan una red, o interrumpen sustancialmente los flujos que se pudieran producir en ésta. Estos actores, que son objeto de ataque y defensa, se pueden descubrir contando cuántas rutas mínimas pasan por ellos; o sea, por su calidad de intermediarios o puntos intermedios.

3.2.3. Detección de comunidades

La detección de comunidades, grupos, cliques (grupos exclusivos), etc. es tema de alto interés en redes sociales. El asunto es complicado pues no es fácil definir un grupo. La definición es fácil cuando hablamos de una estructura formal, cuando existe un grupo definido y un grupo de adherentes que dice ser parte

del grupo. Por ejemplo, Chile y los chilenos. Pero todo se vuelve complicado, oscuro, hasta esotérico cuando hablamos de la estructura informal.

3.2.3.1. Técnicas de detección

Técnicas para detectar grupos hay muchas; hay muchos algoritmos, con muchas velocidades diferentes, que obedecen a diferentes ideas de cómo se detecta un grupo, situación muy opuesta a la de las medidas de centralidad. Una manera tradicional consiste en reducir la detección de grupos a una clasificación o clustering. Dentro de estas técnicas están el tradicional k-Means, los algoritmos genéticos, el análisis de modularidad (el número de vínculos entre gruposes pequeño, dentro de grupos es alto), etc. Adicionalmente, estas técnicas son parametrizables (i.e. número de clases en k-Means, modularidad mínima, etc.), lo que permite analizar la calidad de la clasificación. Aquí se hace posible usar árboles jerárquicos para decidir cuándo la clasificación es buena.

Otra manera tradicional consiste en ver el problema como uno de teoría de grafos. Por ejemplo, la coloración es una forma tradicional de hacer clasificación en grafos. En este caso, también es posible ver el problema como uno de equivalencia estructural transformado a uno de equivalencia regular: “en un grupo de amigos, los amigos compartimos los mismos amigos” (esto define un algoritmo iterativo). Adicionalmente, se pueden buscar cliques y k-Cliques para encontrar los grupos.

3.2.4. Visualización

La visualización de las redes sociales también sirve como método para descubrir propiedades de ésta, aunque tiene menos peso teórico en el análisis. Pero tiene la ventaja de alimentar rápidamente la intuición del investigador. Visualizar redes complejas es un gran desafío; por lo general, se busca presentar gran cantidad de información de forma estética. Se busca la claridad y la simpleza, pese a la gran complejidad de los datos. Y hay que considerar que hay muchas potenciales vistas de los datos, que pueden ilustrar propiedades diferentes: centralidad, comunidades, jugadores clave (que, si desaparecen, desconectan la red), etc. Tal como en la detección de comunidades, existe una gran variedad de algoritmos para visualizar redes sociales. Cada uno obedece a una idea u objetivo diferente, aunque muchas veces se busca la presentación instantánea.

CAPÍTULO 4

Problema

4.1. Planteamiento del problema

El análisis de las redes sociales ha sido una técnica utilizada por empresas desde hace décadas, iniciando con los periódicos y medios impresos. Esto no solo permite conocer a fondo una marca sino también a la competencia y a la industria entera. En la actualidad, no es ningún secreto que las organizaciones que están a la delantera planean sus campañas y estrategias de comunicación a partir de los datos que analizan en línea, lo que les permite conocer a fondo sus clientes y ofrecer campañas y productos altamente especializados.

Estudiar las relaciones entre usuarios se pueden analizar muchas cosas, por ejemplo:

- La velocidad o tiempo con que se propaga una información o tendencia.
- Identificar comunidades.
- Identificar relaciones de algún tipo entre los usuarios.
- Identificar localización.

Las redes sociales están compuestas por actores y vínculos, por lo tanto es intuitivo estructurarlas como un grafo si se definen los actores como vértices y las aristas como los vínculos. En base a una estructura de grafo, realizar análisis es relativamente sencillo y permite aplicar conceptos de la teoría de grafos, la cual es una disciplina que posee muchos avances y tiene años siendo estudiada. El problema es estructurar una red social con muchos usuarios mediante un grafo, es decir con muchos nodos, y que el comportamiento de las relaciones entre ellos sea una distribución similar a una ley de potencias (un usuario puede tener muchas asociaciones con muchos usuarios, y a su vez estos con otros usuarios, y así sucesivamente). En base a estas condiciones, claramente se obtiene un grafo de gran tamaño, en otras palabras es un problema de grandes volúmenes de datos, y utilizar las herramientas tradicionales para almacenar, transformar, procesar,

visualizar y analizar un grafo con estas características, no es posible. Resolver este problema de grandes volúmenes de datos, genera una serie de preguntas, ¿Cuáles son las alternativas de almacenamiento, ventajas y desventajas de cada una?, una vez almacenada ¿Cómo acceder, procesar y generar métricas en base al grafo almacenado?, y finalmente, ¿Cómo visualizar y realizar análisis sobre el grafo procesado?, las respuestas a estas preguntas serán el objeto de estudio del trabajo especial de grado.

4.2. Justificación del problema

4.2.1. ¿Por qué es un problema?

Es un problema ya que es complicado manejar y analizar tanto volumen de datos utilizando herramientas tradicionales para el análisis de datos.

4.2.2. ¿Para quién es un problema?

Es un posible problema para cualquier usuario que le interese y/o necesite realizar un análisis de redes sociales sobre un gran volumen de datos.

4.2.3. ¿Desde cuándo es un problema?

Es un problema desde que existen las redes sociales con muchos usuarios, debido a la gran cantidad de datos que se manejan.

4.3. Objetivos de la investigación

4.3.1. Objetivo general

Realizar análisis estructural sobre una red social a partir de grandes volúmenes de datos, utilizando herramientas de procesamiento paralelo y distribuido.

4.3.2. Objetivos específicos

- Evaluar las herramientas para el procesamiento paralelo y distribuido sobre grandes volúmenes de datos.
- Definir un caso de estudio.
- Implementar una aplicación que permita analizar la estructura de una red social para facilitar la obtención de métricas.
- Desarrollar una interfaz gráfica de usuario que permita visualizar las métricas.
- Realizar pruebas de la aplicación.

Bibliografía

- [1] Dato. <http://definicion.de/datos/>.
- [2] Información. <http://www.definicionabc.com/tecnologia/informacion.php>.
- [3] Conocimiento. <http://sobreconceptos.com/conocimiento>.
- [4] Data Science. <https://datascience.berkeley.edu/about/what-is-data-science/>.
- [5] Big data. <http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data>.
- [6] 7v. <http://www.obs-edu.com/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet/>.
- [7] Base de Datos. <http://searchsqlserver.techtarget.com/definition/database>.
- [8] Base de datos relacionales. <http://searchdatacenter.techtarget.com/es/definicion/Base-de-datos-relacional>.
- [9] Base de datos no relacionales. <http://nosql-database.org/>.
- [10] Almacén de Datos. http://www.sinnexus.com/business_intelligence/datawarehouse.aspx.
- [11] Inteligencia de Negocios. <http://searchdatamanagement.techtarget.com/definition/business-intelligence>.
- [12] Lenguajes de programación. <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
- [13] R. <https://www.r-project.org/>.
- [14] RStudio. <https://www.rstudio.com/>.
- [15] Java. <http://searchsoa.techtarget.com/definition/Java>.
- [16] Python. <http://searchenterpriselinux.techtarget.com/definition/Python>.

- [17] Javascript. <https://www.javascript.com/>.
- [18] Node.js. <https://nodejs.org/en/>.
- [19] Angular.js. <https://angularjs.org/>.
- [20] D3.js. <http://d3js.org/>.
- [21] Apache Hadoop. <https://hadoop.apache.org/>.
- [22] Commons. <https://commons.apache.org/>.
- [23] MapReduce. <https://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce>.
- [24] Hdfs. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
- [25] Yarn. <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>.
- [26] Spark. <http://spark.apache.org/>.
- [27] Accumulo. <https://accumulo.apache.org/>.
- [28] HBase. <http://hbase.apache.org/>.
- [29] Hive. <http://hive.apache.org/>.
- [30] Storm. <http://storm.apache.org/>.
- [31] Mahout. <http://mahout.apache.org/>.
- [32] Falcon. <https://falcon.apache.org/>.
- [33] Flume. <https://flume.apache.org/>.
- [34] Sqoop. <http://sqoop.apache.org/>.
- [35] Knox. <https://knox.apache.org/>.
- [36] Ranger. <http://ranger.apache.org/>.
- [37] Ambari. <https://ambari.apache.org/>.
- [38] Zookeeper. <http://zookeeper.apache.org>.
- [39] Oozie. <http://oozie.apache.org/>.
- [40] Cloudera. <http://www.cloudera.com/content/www/en-us.html>.
- [41] MapR. <https://www.mapr.com/>.
- [42] Hortonworks. <http://hortonworks.com/>.
- [43] Red Social. <http://www.definicionabc.com/social/red-social.php>.

- [44] Redes sociales en Venezuela. <http://www.hinterlaces.com/monitor-pais/80-de-los-venezolanos-cuenta-con-servicio-de-tv-por-suscripcion>.
- [45] Social Digital en Venezuela. <http://www.slideshare.net/wearesocialsg/social-digital-mobile-in-the-americas>.