Disertación
Master en Ingeniería Informática - Computación Móvil

Análisis de Big Data en IoT para campos de Cadenas de Suministro Inteligentes

Fabián Vinicio Constante Nicolalde (2162316)

Leiria, Julio de 2018

Disertación
Master en Ingeniería Informática - Computación Móvil

Análisis de Big Data en IoT para campos de Cadenas de Suministro Inteligentes

Fabián Vinicio Constante Nicolalde (2162316)

Disertación de master realizada bajo la orientación del Doctor António Manuel de Jesús Pereira y del Doctor Fernando José Mateus da Silva, profesores de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria. Co-dirigido por el Msc. Boris Herrera, profesor de la Universidad Central del Ecuador.

Leiria, Julio de 2018

Esta página se ha dejado intencionadamente en blanco

Dedicatoria

Dedico este trabajo de tesis:

A Dios y a la Virgen María que han bendecido mi camino con su presencia, por darme la vida, la sabiduría, ser mí guía en todos los momentos de mi vida y sobre todo la salud para lograr mis objetivos, además de su infinita bondad y amor.

A mi madre, por haberme apoyado en todos los momentos de mi vida, por ser la mejor maestra de mi vida, por haberme educado con buenos sentimientos y valores con sus consejos, por la motivación constante que me ha permitido ser una persona de bien, y sobre todo por brindarme ese inmenso amor de madre.

Al "Instituto de Fomento al Talento Humano" y SENECYT de Ecuador por darme la oportunidad de viajar a Portugal y estudiar en el Instituto Politécnico de Leiria, para mi mejora personal como profesional, buscando que los nuevos conocimientos adquiridos sean una contribución para el beneficio de mi país Ecuador.

Fabián Vinicio Constante Nicolalde.

Esta página se ha dejado intencionadamente en blanco

Agradecimientos

Primero que todo, agradezco a Dios por sus múltiples bendiciones, por todo lo que me ha dado a lo largo de mi vida y por permitirme seguir cumpliendo mis objetivos tanto personal como profesionalmente.

De todo corazón agradezco a mis padres por ser mis guías y motivación en mi vida, de manera especial a mi madre por su incondicional apoyo, consejos y sobre todo el amor que me da día a día. Espero recompensar todo lo que han sabido brindarme.

Quedo totalmente agradecido a la Universidad Central del Ecuador UCE, que me forjó como ingeniero y ahora me permite dar el siguiente paso en mi carrera profesional.

Gracias a todos mis profesores del Instituto Politécnico de Leiria Portugal en especial, a mis tutores, PhD. Antonio Pereira y PhD. Fernando Silva; quienes, con su esfuerzo, conocimiento, motivación, experiencia y dedicación, supieron guiarme en la investigación y desarrollo de mi trabajo de Disertación.

Mi agradecimiento a todas las personas que forman parte de mi vida que de una u otra forma me han brindado su valiosa amistad.

Doy mi agradecimiento al “Instituto de Fomento al Talento Humano" y "SENECYT" de Ecuador por financiar mis estudios de maestría y darme la oportunidad de estudiar en esta prestigiosa institución.

Fabián Vinicio Constante Nicolalde.

Esta página se ha dejado intencionadamente en blanco

Nota Previa

La presente disertación se llevó a cabo en el Centro de Investigación en Informática y Comunicación (CIIC) del Instituto Politécnico de Leiria y en la Universidad Central del Ecuador. Como resultado del trabajo realizado, se produjeron los siguientes artículos:

- Fabián Constante Nicolalde, Fernando Silva, Boris Herrera, António Pereira. "Big Data Analytics in IoT: Challenges, Open Research Issues and Tools". WorldCist'18-6th World Conference on Information Systems and Technologies, Naples – Italy. Publicado el 28 de Marzo de 2018.
- Fabián Constante Nicolalde, Fernando Silva, Boris Herrera, António Pereira. "Herramientas de Análisis de Big Data en IoT y sus Desafíos en Investigaciones Abiertas". CISTI'2018 13th Iberian Conference on Information Systems and Technologies. IEEE, Cáceres – Spain. Publicado el 15 de Junio de 2018.

Esta página se ha dejado intencionadamente en blanco

Resumen

Desde la última década se han producido cantidades voluminosas de datos a medida que aumenta la miniaturización de los dispositivos de Internet de las cosas (IoT). Terabytes de datos se generan día a día a partir de Sistemas de Información Modernos, Computación en la Nube y tecnologías digitales, a medida que crece el número de dispositivos conectados a Internet. Sin embargo, tales datos no son útiles sin poder analítico.

No obstante, el análisis de estos datos masivos requiere muchos esfuerzos en múltiples niveles para la extracción de conocimiento y la toma de decisiones. Por lo tanto, "Análisis de *Big Data*" es un área actual de investigación y desarrollo que se ha vuelto cada vez más importante. Numerosas soluciones de análisis de *Big Data* e IoT, han permitido a la gente obtener información valiosa, aunque estas soluciones están todavía en sus inicios.

Actualmente existe una cierta complejidad involucrada en *Big Data* para superar esto, los ingenieros de software hoy en día empiezan a pensar en *Small Data* ya que combina datos estructurados y no estructurados que pueden medirse en Gigabytes, Peta bytes o Exabytes, siendo parte de pequeños conjuntos de atributos específicos de IoT.

En esta disertación se indagan los esfuerzos de investigación dirigidos al análisis de datos generados por IoT y sistemas transaccionales. Se explica la relación entre el Análisis de *Big Data* e IoT agregando valor al proponer una nueva arquitectura para el análisis de estos datos y un protocolo a seguir para la extracción de conocimiento. Además, se discuten tipos, métodos y tecnologías analíticas para la minería de *Big* IoT *Data*. También se presentan casos de uso notables, desafíos de investigación abiertos como privacidad, visualización e integración de datos y oportunidades que brinda el análisis de datos en el paradigma de IoT.

El trabajo es aplicado al caso de uso específico de "Cadenas de Suministro Inteligente", presentando como una solución propuesta, el análisis de los datos generados desde una "Plataforma de Compra-Venta y Control de Stocks de Productos" que incluye tecnologías RFID (Identificación por Radiofrecuencia) y NFC (Comunicación de Campo Cercano).

Se realiza la gestión casi en tiempo real de transacciones involucradas al manejo de Suministro en una compañía, con el fin de poder analizar *Big Data* generada por estas tecnologías, manejando herramientas de análisis *open source* y poder realizar mejores predicciones y toma de decisiones.

Palabras Claves:

Analisis de Big Data, Internet de las Cosas, Cadenas de Suministro Inteligentes, Minería de Datos, Hadoop, Identificación por Radiofrecuencia.

Esta página se ha dejado intencionadamente en blanco

Abstract

Since the last decade, voluminous amounts of data have been produced as the miniaturization of Internet of Things (IoT) devices increases. Terabytes of data are generated day by day from Modern Information Systems, Cloud Computing and digital technologies, as the number of devices connected to the Internet grows. However, such data is not useful without analytical power.

However, the analysis of these massive data requires many efforts at multiple levels for the extraction of knowledge and decision making. Therefore, "Big Data Analysis" is a current area of research and development that has become increasingly important. Numerous Big Data and IoT analysis solutions have allowed people to obtain valuable information, although these solutions are still in their infancy.

Currently there is a certain complexity involved in Big Data to overcome this, software engineers today begin to think of Small Data as it combines structured and unstructured data that can be measured in Gigabytes, Peta bytes or Exabytes, being part of small sets of specific attributes of IoT.

In this dissertation, research efforts are directed to the analysis of data generated by IoT and transactional systems. The relationship between Big Data Analysis and IoT is explained, adding value by proposing a new architecture for the analysis of this data and a protocol to follow for the extraction of knowledge. In addition, types, methods and analytical technologies for Big IoT Data mining are discussed. There are also notable cases of use, open research challenges such as privacy, visualization and integration of data and opportunities provided by the analysis of data in the IoT paradigm.

The work is applied to the specific use case of "Smart Supply Chains", presenting as a proposed solution, the analysis of the data generated from a "Platform of Purchase-Sale and Control of Stocks of Products" that includes RFID technologies (Radio Frequency Identification) and NFC (Near Field Communication).

The management is carried out almost in real time of transactions involved in the supply management in a company, in order to analyze Big Data generated by these technologies, managing open source analysis tools and be able to perform better predictions and decision making.

Keywords:

Big Data Analytics, Internet of Things, Smart Supply Chains, Data Mining, Hadoop, Radio Frequency Identification.

Esta página se ha dejado intencionadamente en blanco

Índice de Figuras

Índice de Tablas

Esta página se ha dejado intencionadamente en blanco

Lista de Acrónimos

Acrónimo	Designación
AUC	Área bajo la curva ROC
BDA	*Big Data Analytics*
CDH	*Cloudera Distributed Hadoop*
CRM	*Customer Relationship Management*
DW	*Data Warehouse*
EDW	*Enterprise Data Warehouse System*
ERP	*Enterprise Resource Planning*
ETL	*Extract, Transform and Load*
HDFS	*Hadoop Distributed File System*
HTML	*HyperText Markup Language*
IBDA	*IoT Big Data Analytics*
IoT	*Internet of Things*
JDBC	*Java Database Connectivity*
KNN	*k-Nearest Neighbors*
MPP	*Massively Parallel Processing*
NGTS	*Next Generation Technologies and Services*
ODBC	*Open DataBase Connectivity*
PNL	Procesamiento de Lenguaje Natural
POS	*Point of Sale*
RDD	*Resilient Distributed Datasets*
RCFile	*Record Columnar File*
RFID	*Radio Frequency Identification*
ROC	*Receiver Operating Characteristic*
RPART	*Recursive Partitioning and Regression Trees*
SKU	*Stock Keeping Unit*
SQL	*Structured Query Language*
SVM	*Support Vector Machine*
TCP	Protocolo de Control de Transmisión

Índice

1. Introducción

El término “Internet de las Cosas” (IoT) es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet, permitiendo recopilar e intercambiar datos de manera constante, cubriendo un gran número de protocolos, arquitecturas, estándares y aplicaciones, para la adquisición de datos ubicuos y análisis a gran escala siendo un gran campo de investigación y análisis [1].

Gartner espera que para el año 2020, 26 mil millones de objetos estén conectados a Internet y, por lo tanto, prediciendo una revolución digital, superando varias barreras tecnológicas: desde la necesidad de abordar de forma única (IPv6) a la necesidad de alimentarlo o recargarlo (baterías innovadoras, cosechadoras de energía futurista, técnicas y tecnologías de generación y gestión). Los dispositivos y objetos están interconectados a través de una variedad de soluciones de comunicación, como *Bluetooth, Wifi, ZigBee* y GSM. Estos dispositivos de comunicación transmiten datos y reciben comandos de dispositivos controlados remotamente, que permiten la integración directa con el mundo físico a través de sistemas informáticos mejorando así los niveles de vida [1].

Una gran cantidad de datos se generan cada día y *Big Data* es una frase que se utiliza para describir este fenómeno. ESG Re-search (2012) indica que *Big Data* es una recopilación de un gran volumen de datos (tanto estructurados como no estructurados) lo suficientemente grandes para que las bases de datos y las técnicas de software tradicionales no sean útiles en caso de procesamiento [2].

Los sistemas de bases de datos tradicionales son ineficientes para almacenar, procesar y analizar la creciente cantidad de datos. Esto caracteriza a *Big Data* en tres aspectos: (a) fuentes de datos, (b) análisis de datos, y (c) presentación de resultados analíticos. Esta definición utiliza el modelo 3V (volumen, variedad, velocidad) propuesto por Beyer [1].

Las soluciones de *Big Data* que respaldan la planificación empresarial integrada actualmente ayudan a las organizaciones a organizar cadenas de suministro más receptivas a medida que comprenden mejor las tendencias del mercado y las preferencias de los clientes al evitar retrasos en las entregas mediante el análisis de los datos de GPS además del tráfico y los datos meteorológicos para planificar dinámicamente y optimizar las rutas de entrega [3].

Big Data e IoT están surgiendo en una escala tan masiva y formándose entre sí en 2020, habría alrededor de 47 Zettabytes de datos creados que son 300 veces más que 2005, cuanto más IoT crece, más demandas se colocan en las capacidades de *Big Data*, y viceversa. Las tecnologías tradicionales de almacenamiento de datos, por ejemplo, ya están siendo llevadas a sus límites, lo que lleva a soluciones más innovadoras y avances en la tecnología para manejar cargas de trabajo cada vez mayores. Se prevé que más de 50.000 millones de dispositivos que van desde teléfonos inteligentes, portátiles, sensores y consolas de juegos estarán conectados a Internet a través de varias redes heterogéneas de acceso [4]. Habilitado

por tecnologías, como la Identificación por Radiofrecuencia (RFID) y redes de sensores inalámbricos.

1.1. Identificación del problema

IoT surgió en base a la necesidad de las Cadenas de Suministro y la identificación de objetos, personas y animales mediante el uso de etiquetas inteligentes RFID. Con ellas se consiguió otorgar de un identificador único al objeto deseado. Para la existencia de IoT, son necesarias tres cosas: inteligencia integrada en los objetos, la conectividad de los objetos a Internet y la interacción entre los propios objetos [5].

Existen en la actualidad grandes cantidades voluminosas de datos, desde la última década con la miniaturización de Internet cosas (IoT) los dispositivos han ido aumentando. Sin embargo, estos datos obtenidos no son útiles sin poder analítico que permita realizar una buena toma de decisiones.

Big Data, Small Data, IoT y soluciones de análisis han permitido a las personas obtener información valiosa sobre los grandes datos generados por los dispositivos IoT; estas soluciones están todavía en su infancia, y el dominio carece de una encuesta exhaustiva.

1.2. Interrogantes de la investigación

Las interrogantes que se ha planteado al realizar el trabajo de investigación son las siguientes:

1) ¿Cómo se pueden utilizar herramientas de Análisis de *Big Data* e *IoT* para optimizar el proceso de registro y control de productos en Cadenas de Suministro?

2) ¿De qué manera se puede analizar un gran volumen de datos que no pueden aprovecharse utilizando herramientas tradicionales?

3) ¿Cómo se puede saber las tendencias más actuales de compra, determinar si una transacción es o no fraudulenta y la influencia en los usuarios?

4) ¿Es posible aplicar modelos de Minería de Datos o *Machine Learning*, para poder realizar un análisis que permita a los administradores de la Cadena de Suministro mejorar el envío automatizado?

1.3. Justificación e importancia

La investigación va dirigida con el fin de proponer una nueva arquitectura que permita la explicación y análisis de los datos IoT generados. Además, se discuten los tipos, métodos y

tecnologías analíticas de *Big Data* para la Minería de Datos de gran tamaño. También se presentan numerosos casos de uso notables, se analizan varias oportunidades proporcionadas por el análisis de datos en el paradigma IoT, se presentan desafíos de investigación abiertos como la privacidad, la minería de datos, la visualización y la integración.

Las directrices de esta investigación; se aplican al caso específico de "Cadenas De Suministro Inteligente", presentando como una solución propuesta, el análisis de los datos generados desde una "Plataforma de Compra-Venta y Control de Stocks de Productos" que incluye tecnologías RFID y NFC para la gestión casi en tiempo real de la rotación de inventarios y transacciones relacionados con el manejo de Suministro en una empresa.

La información recopilada por estas tecnologías proporciona una visibilidad detallada de los artículos enviados desde el fabricante a un minorista; permitiendo a los administradores de la Cadena de Suministro predecir y tomar decisiones factibles para la organización.

1.4. Objetivos

La Cadena de Suministro se está convirtiendo en una plataforma de comunicación por la que el uso de tecnologías con sensores IoT, *Big Data* aporta avances de vanguardia y nuevas oportunidades de negocio relacionadas con la experiencia del cliente a través de todos los canales de venta y dispositivos. A continuación, se define el objetivo general y objetivos específicos de la disertación.

1.4.1. Objetivo General

Analizar los datos obtenidos de los dispositivos IoT y de una plataforma altamente transaccional, enfocados en el Control Inteligente de las Cadenas de Suministros, utilizando herramientas de Análisis de *Big Data* y herramientas enfocadas a la Minería de Datos, con el fin de poder realizar predicciones y una mejor toma de decisiones.

1.4.2. Objetivos Específicos

Los objetivos específicos de esta investigación son:

- Realizar un estudio de la influencia y relación existente de *Big Data Analytics* y *Small Data* en IoT.
- Analizar oportunidades proporcionadas por el análisis de datos dentro del paradigma de IoT, se presentan desafíos de investigación abiertos como: Privacidad, Minería de Datos, modelos de *Machine Learning*, la Visualización de Datos y la Integración.
- Estudio de aplicaciones de *Big Data* y *Small Data* en escenarios basados en IoT.
- Proponer una nueva arquitectura para el análisis de datos IoT, en la que se discuten los tipos, métodos y tecnologías analíticas de *Big Data* IoT para la Minería de Datos de gran tamaño.

- Analizar *Big Data* generada por las tecnologías RFID y NFC utilizadas por una "Plataforma de Compra-Venta y Control de Stocks de Productos", utilizando herramientas de análisis de *Big Data open source*.
- Proponer un protocolo para obtener conocimiento en base al análisis de *Big Data*.

1.5. Metodología de la Investigación

La metodología de investigación utilizada para alcanzar los principales objetivos de este trabajo, incluye varias fases, de alguna forma evidenciadas por la lista de objetivos a alcanzar.

En una primera fase, se procede a la investigación y recogida de la literatura referente a desarrollos identificados como relevantes en cada una de las áreas del trabajo. El camino a recorrer en la fase de concepción y caracterización de la solución propuesta.

En una segunda fase, se realizan reuniones con los orientadores del IPL; el objetivo primordial fue el de recolectar datos de los sensores RFID y de información transaccional sobre manejo de Suministro y necesidades al que la arquitectura propuesta para el análisis de estos datos debe responder.

En una tercera fase, se ejecutan algunas pruebas, con el objetivo de conocer y manejar las herramientas de procesamiento de *Big Data* e identificar en base a los resultados: riesgos, problemas y tendencias.

La cuarta fase, se basa en el conocimiento científico adquirido y las pruebas realizadas en la fase anterior, se ha concebido y caracterizado una propuesta de arquitectura y mecanismos de procesamiento y análisis de datos para dar respuesta a las necesidades y problemas identificados.

La quinta y última fase, se obtiene conclusiones, se generan aportes sobre el trabajo realizado y se propone trabajos futuros dentro de esta línea de investigación.

1.6. Estructura de la Disertación

Esta disertación se encuentra estructurada en seis capítulos, que reflejan el trabajo desarrollado para alcanzar los objetivos anteriormente presentados.

En el presente capítulo se muestra la identificación del problema a abordar, interrogantes de la investigación, justificación e importancia de esta tesis y se definen los principales objetivos del trabajo.

En el Capítulo 2, se realiza el levantamiento del Estado de Arte del conocimiento científico en el área de *Big Data*, *Small Data* y IoT, siendo identificados los principales conceptos,

modelos y tendencias. Posteriormente, en el Capítulo 3, se mencionan las herramientas *open source*, existentes más importantes para el Análisis de *Big Data* en IoT.

El caso de estudio, así como el análisis de la arquitectura planteada y el flujo de trabajo de *Big Data* enfocado a Cadenas de Suministro Inteligente, se caracterizan en el Capítulo 4, siendo también presentado la solución práctica propuesta.

En el capítulo 5 se analizan los datos en base a la aplicación de Minería de Datos y modelos de *Machine Learning* que mejor se adapten a las necesidades de los usuarios, la validación de las funcionalidades de las herramientas de procesamiento utilizadas, escalabilidad, rendimiento, usabilidad, simplicidad y generación de conocimiento con los datos analizados.

Las conclusiones y planteamientos de trabajos futuros de la disertación, se efectúan en el Capítulo 6, concluyendo el trabajo de investigación y cumpliendo con los objetivos inicialmente propuestos.

Esta página fue intencionalmente dejada en blanco

2. Estado del Arte

En este apartado se realiza una investigación detallada de los conceptos generales, desafíos, la influencia de *Big Data* en IoT, las principales técnicas para el Análisis de *Big Data* en IoT, así como de aquellos trabajos que por tener un fin similar al de esta tesis, sirven como base de discusión para desarrollar una solución al problema a resolver.

2.1. Big Data

Es un término que se refiere a una gran cantidad de datos (estructurados, no estructurados y semiestructurados) que excede la capacidad del software convencional para capturarse, administrarse y procesarse en un tiempo razonable. En el año 2012, se estimó que su tamaño debería ser de entre una docena de Terabytes y varios Petabytes de datos en un solo conjunto de datos [6]. En general, *Big Data* pueden clasificarse en dos categorías: "Estructurado" y "No estructurado"[7]. Se muestra su definición en la Tabla 1.

Tipo	Definición	Ejemplo
Estructurado	Datos que pueden ser identificados inmediatamente dentro de una estructura / base de datos electrónica.	El nombre de una ciudad del campo "ciudad" de un formulario.
No estructurado	Datos que no están en ubicaciones fijas y necesitan ser escaneados y analizados.	Texto libre en documentos, correos electrónicos, blogs, etc.

Tabla 1. Datos estructurados vs. Datos no estructurados [7].

"McKinsey Global Institute" definió *Big Data* como el tamaño de los conjuntos de datos que son una herramienta de sistema de base de datos mejor que las herramientas habituales para capturar, almacenar, procesar y analizar dichos datos [8].

"*The Digital Universe*" escribe sobre tecnologías de datos como una nueva generación de tecnologías y arquitecturas que intentan extraer el valor de un volumen masivo de datos en varios formatos permitiendo la captura, el descubrimiento y el análisis de alta velocidad [9]. Este estudio previo también caracteriza *Big Data* en tres aspectos: (a) fuentes de datos, (b) análisis de datos y (c) presentación de resultados de análisis.

Beyer define un modelo que utiliza los 3V (volumen, variedad, velocidad) para describir *Big Data* [10]. El volumen se refiere a la enorme cantidad de datos que se generan diariamente, mientras que la velocidad es la tasa de crecimiento y la rapidez con que se recopilan los datos para el análisis. La variedad proporciona información sobre los tipos de datos, como estructurados, no estructurados y semiestructurados [10]. La Figura 1 , se refiere a esta última definición de *Big Data.*

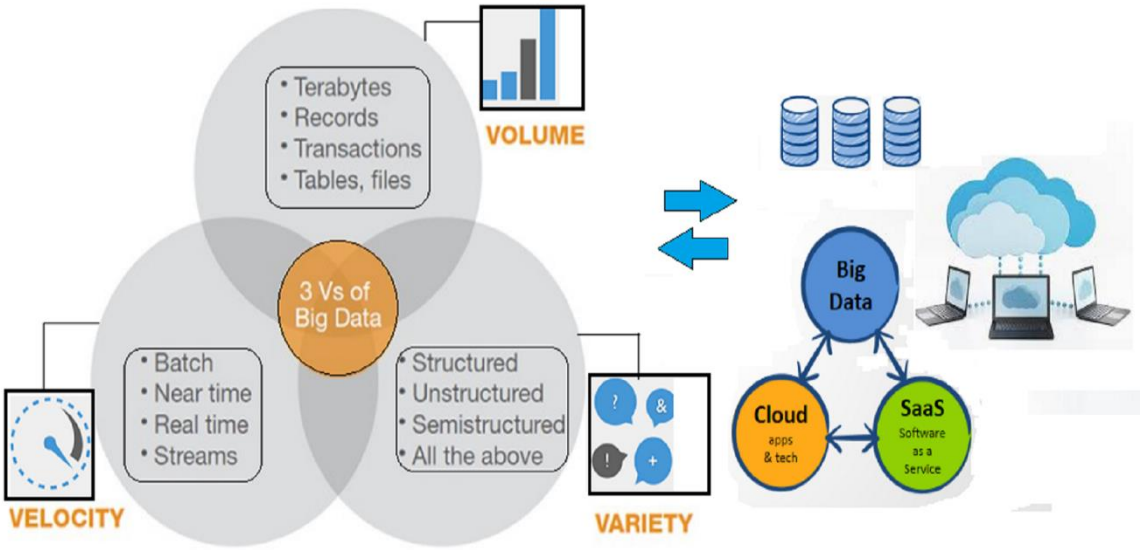


Figura 1. Definición de *Big Data* como los tres vs. Adaptado de [10].

2.2. Big Data Analytics (BDA)

BDA es el proceso de examinar grandes conjuntos de datos que contienen una variedad de tipos de datos [11] para revelar patrones invisibles, las correlaciones ocultas, las tendencias del mercado, las preferencias de los clientes, y otra información útil de negocios[12]. Implica los procesos de búsqueda en una base de datos, minería y análisis de datos, con el objetivo de mejorar el rendimiento de las organizaciones [12], [13].

BDA surge como una respuesta a estas necesidades, ya que se centra en el estudio para mejorar la capacidad de obtener, almacenar, analizar y visualizar millones de datos que serían inaccesibles a los procesos o herramientas de análisis convencionales [14].

El objetivo principal de BDA es ayudar a cualquier área de investigación a mejorar la comprensión de los datos y, por lo tanto, a tomar decisiones eficientes e informadas. BDA permite analizar un gran volumen de datos que no pueden ser explotados con herramientas tradicionales [12].

BDA requiere tecnologías y herramientas que puedan transformar una gran cantidad de datos estructurados, no estructurados y semiestructurados en un formato más completo de datos y metadatos para procesos analíticos. Los algoritmos utilizados en estas herramientas analíticas deben descubrir patrones, tendencias y correlaciones en una variedad de horizontes de tiempo en los datos. Después de analizar los datos, estas herramientas visualizan los hallazgos en tablas, gráficos y gráficos espaciales para una toma de decisiones eficiente.

El desafío se centra en el rendimiento de los algoritmos actuales utilizados en BDA, que no aumenta linealmente con el rápido aumento de recursos computacionales [15]. La Tabla 2 , muestra los Sistemas Analíticos existentes.

Tipos de Análisis	Uso Especifico	Arquitecturas existentes / Herramientas	Ventajas
Tiempo Real	Para análisis de grandes cantidades de datos generados por Sensores	Greenplum HANA	Clústeres de procesamiento paralelo utilizando bases de datos tradicionales, plataformas de computación basadas en memoria
Offline	No hay altas exigencias en tiempo de respuesta	Scribe Kafka Timetunnel Chukwa	Eficiente adquisición de datos Reducción de costo de la conversión de formato de datos
Nivel de memoria	Para usar cuando el volumen total de datos es menor que el máximo Memoria del cluster	MongoDB	Tiempo Real
Nivel de *Business intelligence*	Para usar cuando la escala de datos sobrepasa el nivel de memoria	Data analysis plans	Ambas offline y online
Nivel Masivo	Cuando la escala de datos es totalmente superior a la capacidad de los productos de inteligencia de negocios y bases de datos tradicionales.	MapReduce	Principalmente pertenece a Offline

Tabla 2. Análisis de Sistemas Analíticos existentes. Adaptado de [1].

2.2.1. Categorías de los desafíos en BDA

Los desafíos de BDA se clasifican en 4 categorías generales: Almacenamiento y Análisis de Datos, Descubrimiento del Conocimiento y Complejidades Computacionales; Escalabilidad y Visualización de Datos; y la Seguridad de la Información [16].

2.2.1.1. Almacenamiento y Análisis de Datos

Tiene un elevado costo, por tanto, el primer reto de BDA es el medio de almacenamiento y una mayor velocidad de entrada / salida. En este caso, la accesibilidad de los datos debe priorizarse para el descubrimiento y la representación del conocimiento. En décadas pasadas, el analista utilizaba unidades de disco duro para almacenar datos, pero su rendimiento de entrada / salida aleatoria es más lenta que el de entradas / salidas secuenciales. Sin embargo, las tecnologías de almacenamiento disponibles no pueden tener el rendimiento requerido para procesar *Big Data* [16].

Otro desafío con el análisis de *Big Data* se atribuye a la diversidad de datos. La reducción de datos, la selección de datos y la selección de características son tareas importantes, debido al gran tamaño de los conjuntos de datos [16]. Esto sucede porque los algoritmos existentes no siempre responden en un momento apropiado cuando se trata de datos de alta dimensión. La automatización de este proceso y el desarrollo de nuevos algoritmos de aprendizaje automático para garantizar la coherencia es un gran desafío en los últimos años [17].

Tecnologías recientes, como Hadoop [18] y MapReduce [19], permiten la recopilación de una gran cantidad de datos semiestructurados y no estructurados en un período de tiempo

razonable. El desafío se centra en cómo analizar efectivamente estos datos para obtener un mejor conocimiento. Un marco estándar para analizar datos es transformar datos semiestructurados o no estructurados en datos estructurados, y luego aplicar algoritmos de minería de datos para extraer conocimiento, discutido por Das y Kumar [20].

2.2.1.2. Descubrimiento del Conocimiento y Complejidades Computacionales

Incluye varios campos secundarios, como la autenticación, el *archiving*, la administración, la preservación, la recuperación de información y la representación. Debido al aumento en el tamaño de *Big Data*, las herramientas existentes pueden no ser eficientes para procesar esta información para obtener información significativa. Los enfoques más populares en el caso de la gestión de datos son *Data Warehouse* (DW) y *Datamarts*. Un DW es el principal responsable de almacenar los datos que se obtienen de los sistemas operativos, mientras que un *Datamart* se basa en un DW y facilita el análisis [16].

El objetivo básico de estas investigaciones es minimizar el procesamiento de los costos computacionales y las complejidades, sin embargo, las herramientas de BDA actuales tienen un rendimiento bajo al manejar complejidades computacionales, incertidumbres e inconsistencias. Esto lleva a un gran desafío para desarrollar técnicas y tecnologías que puedan manejar la complejidad computacional, la incertidumbre y las inconsistencias de una manera efectiva [16].

2.2.1.3. Escalabilidad y Visualización de Datos

El desafío más importante para las técnicas de BDA es su escalabilidad y seguridad. En las últimas décadas, los investigadores han prestado atención para acelerar el análisis de datos y acelerar el procesamiento, seguido por la Ley de Moore [16]. La escalabilidad de los datos se ha vuelto necesaria para muchas organizaciones que se ocupan de conjuntos de datos explosivos, precisamente cuando surgen problemas de rendimiento. Una plataforma de datos escalable acomoda los cambios rápidos en el crecimiento de datos, ya sea en tráfico o volumen, usando agregado de hardware o software [21].

El objetivo de la visualización de datos es presentar los datos de una manera más apropiada, utilizando algunas técnicas de teoría gráfica. Los mercados en línea, como Flipkart, Amazon o e-bay, tienen millones de usuarios y miles de millones de productos para vender cada mes, esto genera una gran cantidad de datos. Algunas empresas usan la Tableau para visualizar *Big Data* [16]. Tableau es una plataforma analítica centralizada para el descubrimiento y la exploración de datos que combina los dos activos más importantes de una empresa: su gente y sus datos (tanto *Big Data* como datos de menor volumen) [22].

2.2.1.4. Seguridad de la Información

En BDA, una gran cantidad de datos se correlacionan, analizan y utilizan para extraer patrones significativos. Todas las organizaciones tienen diferentes políticas para proteger su

información confidencial. Preservar información sensible es un problema importante en BDA. Existe un gran riesgo de seguridad de la información que se está convirtiendo en un importante problema de análisis de datos. La seguridad de *Big Data* se puede mejorar mediante el uso de técnicas de autenticación, autorización y encriptación. Se debe prestar atención al desarrollo de un modelo de política de seguridad y un sistema de prevención multinivel [16].

2.2.2. Métodos de BDA

BDA pretende extraer inmediatamente información que ayude a hacer predicciones, identificar tendencias recientes, encontrar información oculta y tomar decisiones. Las técnicas de minería de datos se implementan ampliamente tanto para métodos específicos de problemas como para análisis de datos generalizados. En consecuencia, se utilizan métodos estadísticos y métodos de *Machine Learning*. Aspectos de la gestión de *Big Data* captura, almacenamiento, pre procesamiento y análisis. [23].

La minería de datos juega un papel importante en el análisis, y la mayoría de las técnicas se desarrollan utilizando algoritmos de minería de datos de acuerdo con un escenario en particular [23]. Se presenta varios métodos de análisis que se pueden implementar para varios casos de estudio de *Big Data*. Algunos de estos son eficientes para análisis de *Big Data* en IoT. Los métodos presentados son: Clasificación, *Clustering*, Predicción, Arboles de Decisión y Minería de Reglas de Asociación [23].

La Figura 2 , muestra y resume cada una de estas categorías. Cada categoría es una función de minería de datos e implica varios métodos y algoritmos para cumplir con los requisitos de extracción y análisis de información.

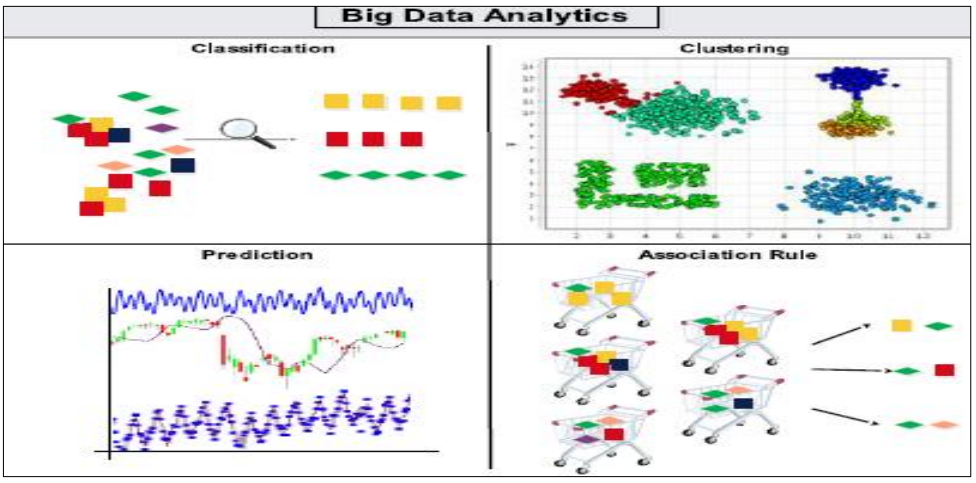


Figura 2. Métodos de BDA. Adaptado de [1].

Métodos de Clasificación son presentandos como, por ejemplo: Red Bayesiana, Máquinas de Soporte Vectorial (SVM) y el vecino K-más cercano (KNN). Del mismo modo, la Partición, la Clusterización Jerárquica, y la Co-ocurrencia son generalizadas en la

Clusterización. La Minería de Reglas de Asociación y la Predicción incluyen métodos significativos.

2.2.2.1. Clasificación

Este método es un enfoque de aprendizaje supervisado que utiliza el conocimiento previo como datos de formación para clasificar los objetos de datos en grupos. Se asigna una categoría predefinida a un objeto y, por lo tanto, se logra el objetivo de predecir un grupo o una clase para un objeto. Encontrar patrones desconocidos u ocultos es una tarea critica para *Big Data* de IoT [24].

La Red Bayesiana es un Método de Clasificación que ofrece un modelo de interoperabilidad. Es eficiente para analizar estructuras de datos complejas reveladas a través de *Big Data* en lugar de formatos de datos estructurados tradicionales. Estas redes se dirigen a los gráficos acíclicos, donde los nodos son variables aleatorias y los bordes denotan la dependencia condicional. Naïve Selectivo, semi-naïve Bayes, Bayes multi-nets son las categorías propuestas para la clasificación [25] .

El análisis de los patrones de datos y la creación de grupos se llevan a cabo eficientemente utilizando SVM. Utiliza la teoría del aprendizaje estadístico para analizar los patrones de datos y crear grupos. Varias aplicaciones de clasificación SVM en el BDA incluyen la clasificación de texto, coincidencia de patrones, diagnósticos de salud, y el comercio[26].

KNN proporciona mecanismos eficaces para encontrar los patrones ocultos de conjuntos de *Big Data*, de modo que los objetos recuperados son similares a la categoría predefinida. La Clasificación es una de las técnicas más extendidas de minería de datos para BDA [26].

2.2.2.2. Análisis de Clúster

El Análisis de Clúster o *Clustering* es otra técnica de Minería de Datos utilizada como un gran método de BDA. Contrariamente a la Clasificación, *Clustering* utiliza un enfoque de aprendizaje sin supervisión y crea grupos para determinados objetos sobre la base de sus características distintivas significativas. Los métodos más utilizados para *Clustering* son: Clúster Jerárquico y Clúster Particional [26].

Clusterización Jerárquica: Combina pequeños clústeres de objetos de datos para formar un árbol jerárquico y crear clústeres aglomerados. El análisis de mercado y la toma de decisiones empresariales son las aplicaciones más importantes de BDA[26].

Clusterización Particional: Su objetivo es obtener una partición de los objetos en grupos o clústeres de tal forma que todos los objetos pertenezcan a alguno de los k clústeres posibles y que por otra parte los clústeres sean disjuntos. Uno de los problemas con los que uno se enfrenta en aplicaciones prácticas es el desconocimiento del valor de k adecuado [27]. Existen algoritmos que son capaces de adaptar el valor de k, a medida que se lleva a cabo la

búsqueda. Otra opción es cortar el dendrograma (gráfico usado en el procedimiento jerárquico que permite visualizar el proceso de agrupamiento del clúster en los distintos pasos, formando un diagrama en árbol) [27].

2.2.2.3. Reglas de Asociación

La Minería de Reglas de Asociación implica la identificación de relaciones interesantes entre diferentes objetos, para análisis de tendencias del mercado, el comportamiento de compra del consumidor y las predicciones de la demanda del producto [26].

2.2.2.4. Análisis Predictivo

Utiliza datos históricos, que son conocidos como datos de entrenamiento, para determinar los resultados como tendencias o comportamiento en los datos. Los algoritmos SVM y de lógica difusa se utilizan para identificar relaciones entre variables independientes y dependientes y para obtener curvas de regresión para predicciones, como, por ejemplo, desastres naturales. Además, las predicciones de compra de los clientes y las tendencias de las redes sociales se analizan a través de análisis predictivos. El análisis de series de tiempo reduce la alta dimensionalidad asociada con *Big Data* y ofrece representación para una mejor toma de decisiones [1].

Los análisis predictivos son útiles para predicciones de desastre y mercado, mientras que el análisis de series de tiempo se usa en pronósticos de desastres, imágenes médicas, reconocimiento de voz, análisis de redes sociales y gobierno electrónico [1].

2.3. Internet de las Cosas (IoT)

IoT proporciona una plataforma para que los sensores y dispositivos se comuniquen sin problemas dentro de un entorno inteligente y permite el intercambio de información entre plataformas de manera conveniente [1]. La reciente adaptación de diferentes tecnologías inalámbricas coloca a IoT como la próxima tecnología revolucionaria para aprovechar todas las oportunidades que ofrecen las tecnologías de Internet [28].

Los dispositivos de recopilación de datos detectan datos y transmiten datos utilizando dispositivos de comunicación integrados a través de una variedad de soluciones de comunicación tales como Bluetooth, WI-Fi, ZigBee y GSM. Estos dispositivos transmiten datos y reciben comandos desde aparatos controlados remotamente que permiten la integración directa con el mundo físico a través de sistemas informáticos para mejorar los niveles de vida [1].

Se espera que más de 50 mil millones de dispositivos, estén conectados a Internet a través de varias redes de acceso heterogéneas habilitadas por tecnologías como la identificación por radiofrecuencia (RFID) y redes de sensores inalámbricos [1]. IoT se reconoce en tres paradigmas: orientado a Internet; sensores; y el conocimiento [29]. A continuación, se presentan las tecnologías más importantes utilizadas en IoT.

2.3.1.1. Radio Frecuency Identification (RFID)

Es una tecnología de punta para la completa identificación de objetos de cualquier tipo que permite una rápida captura de datos de manera automática mediante radio frecuencia [30]. RFID es un método de identificación automática basado en el almacenamiento y recuperación de datos que utilizan ciertos dispositivos llamados etiquetas RFID [31]. Las etiquetas contienen dispositivos receptores y transmisores de señales, que emiten mensajes legibles por los lectores RFID [32].
Un sistema RFID está conformado habitualmente por tres elementos: etiquetas (*Tags*), lectores y Middleware para integrar datos con diferentes aplicaciones [33]. El uso de esta tecnología tiene las siguientes ventajas:

- **Mayor automatización** en el proceso de lectura de las etiquetas, la lectura se puede realizar sin necesidad de tener una línea de visión directa con el dispositivo lector.
- **Ahorro en tiempo** de lectura de las tarjetas ya que es posible realizar la lectura simultanea de más de una etiqueta.
- **Visibilidad** completa de toda la información almacenada dado que la información permanece intacta en la etiqueta [31].

i) Etiquetas RFID

Dotadas de un microchip que almacena datos y un circuito impreso a modo de antena emisora, utilizado para comunicarse a través de las señales de radio frecuencia. Se pueden unir a cualquier artículo, se clasifican en dos categorías generales: activas (tienen su propia fuente de energía, pueden acceder a ellas desde una distancia más lejana (de 20 a 100 metros) y pasivas (no cuentan con batería integrada, recogen la energía del campo electromagnético creado por el lector), dependiendo de su fuente de energía eléctrica [33].

ii) Lectores RFID

Dispositivos electrónicos que se comunican con las etiquetas través de la antena y leen la información almacenada en la etiqueta RFID. El lector puede tener diversas formas de diseño ya sea como una forma fija o como un terminal móvil.

El lector de RFID crea un campo de frecuencia de radio que detecta las ondas y puede ser capaz de leer datos desde un transpondedor (dispositivo transmisor de señales) y escribir datos hacia este [33].

iii) Middleware RFID

Es un tipo especial de software utilizado para recoger y filtrar datos de los dispositivos de lectura RFID. A través de este software se gestiona en tiempo real la información de lectura que han hecho los lectores, se recopilan los datos procesados, se transforman y se transfieren a otros sistemas de información existentes [33].

iv) Código Electrónico de Producto (EPC)

Identificador universal basado en Identificadores Universales de Recursos (URIs); este código proporciona una identidad única para cada objeto físico en cualquier parte del mundo y para todos los tiempos [34].

El EPC es un esquema de numeración que proporciona una identificación única para objetos físicos y sistemas. La numeración en EPC está basada en EPC-64, EPC-96 y EPC-256, tres modos de codificación, que son respectivamente 64, 96 y 256 bits de longitud [34]. El EPC incluye:

- Cabecera que identifica la longitud, tipo, estructura, versión y generación del EPC;
- Número de Administrador que identifica la empresa o fabricante del objeto;
- Clase de objeto;
- Número de serie, que es la instancia especifica de la clase de objeto que se etiqueta.

Las etiquetas de RFID almacenan un EPC único en un chip y transmiten este código a través de una antena para lectores de RFID [35].

2.3.1.2. Near Field Communication (NFC)

Es una tecnología inalámbrica de comunicación entre dispositivos (especialmente teléfonos móviles y asistentes personales). Esta tecnología fue desarrollada por Philips y Sony en 2002, y combina la tecnología de conectividad inalámbrica RFID y tecnologías de interconexión para ofrecer una comunicación inalámbrica de corto alcance y de alta frecuencia entre dos dispositivos NFC ubicados a menos de 20 cm [36].

Los sistemas NFC constan de dos elementos: a) el iniciador, el cual comienza y controla el intercambio de información y b) el objetivo, que es el dispositivo que responde al requisito del iniciador [37].

2.4. Relación entre IoT y BDA

BDA en IoT requiere procesar una gran cantidad de datos sobre la marcha y almacenar los datos en varias tecnologías de almacenamiento. Gran parte de los datos no estructurados se recopilan directamente de las "cosas" habilitadas para la Web, las implementaciones de *Big Data* requieren realizar analíticas rápidas con consultas grandes para permitir a las organizaciones obtener información rápida, tomar decisiones rápidas e interactuar con personas y otros dispositivos [1]. La relación entre IoT y *Big Data* se divide en 3 pasos:

Primer paso: Gestión de fuentes de datos IoT, en las que los dispositivos sensores conectados utilizan aplicaciones para interactuar entre sí, generando grandes cantidades de fuentes de datos con diferentes formatos. Estos datos pueden almacenarse con ayuda de herramientas de bajo coste en la nube [1].

Segundo paso: Generación de datos llamado "*Big Data*", que se basan en su volumen, velocidad y variedad. Estas enormes cantidades de datos se almacenan en grandes bases de datos distribuidas de tolerancia a fallos [1].

Tercer paso: Herramientas de Análisis como MapReduce, Spark, Splunk y Skytree, que pueden analizar los grandes conjuntos de datos de IoT almacenados [1]. La Figura 3, muestra la relación entre IoT y análisis de *Big Data*.

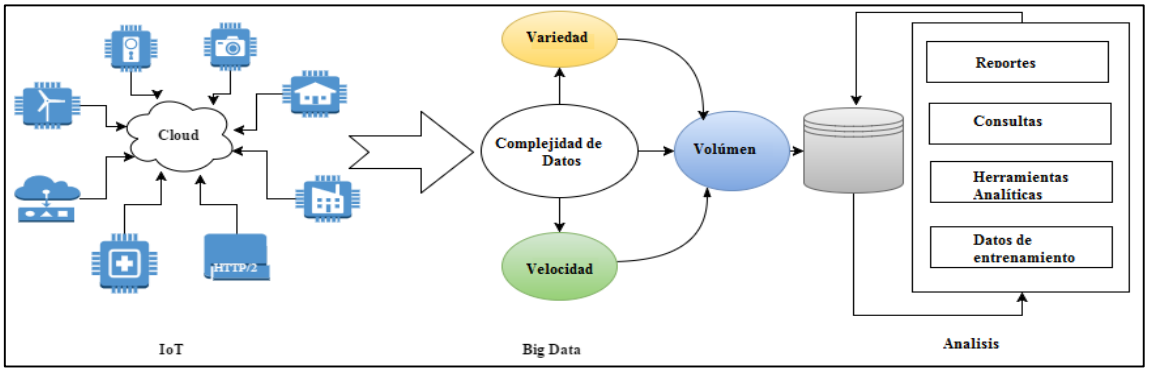


Figura 3. Relación entre IoT y BDA. Adaptado de [1].

2.4.1. Problemas de investigación abierta en IoT para BDA

IoT tiene un impacto económico y social imperativo para la futura construcción de información, redes y tecnologías de comunicación. Presenta desafíos en combinaciones de volumen, velocidad y variedad. Varias tecnologías diversificadas tales como la inteligencia computacional y los datos de gran tamaño se pueden incorporar para mejorar la gestión de datos y el descubrimiento del conocimiento de las aplicaciones de automatización a gran escala [38]. La Figura 4 muestra una descripción general del proceso de descubrimiento de datos y conocimiento en IoT.

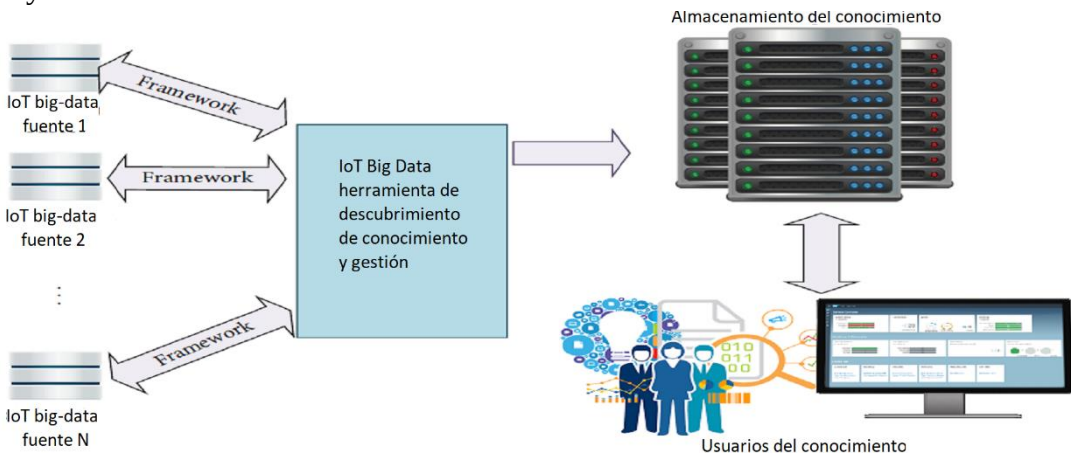


Figura 4. Descubrimiento del conocimiento de *Big Data* en IOT. Adaptado de [38].

El mayor desafío presentado por *Big Data* es la adquisición de conocimiento a partir de datos de IoT. Es esencial desarrollar infraestructuras para analizar los datos de IoT. Numerosos dispositivos de IoT generan flujos continuos de datos y los investigadores pueden desarrollar herramientas para extraer información significativa a partir de estos datos utilizando técnicas de aprendizaje automatizadas [38].

Comprender los flujos de datos y analizarlos para obtener información significativa es un desafío y lleva a BDA. Los algoritmos de aprendizaje automático y las técnicas de inteligencia computacional son la única solución para manejos de estos datos [31]. El sistema de exploración del conocimiento, ilustrado en la Figura 5, consta de 4 segmentos: adquisición de conocimiento; base de conocimientos; difusión del conocimiento y aplicación del conocimiento [38].

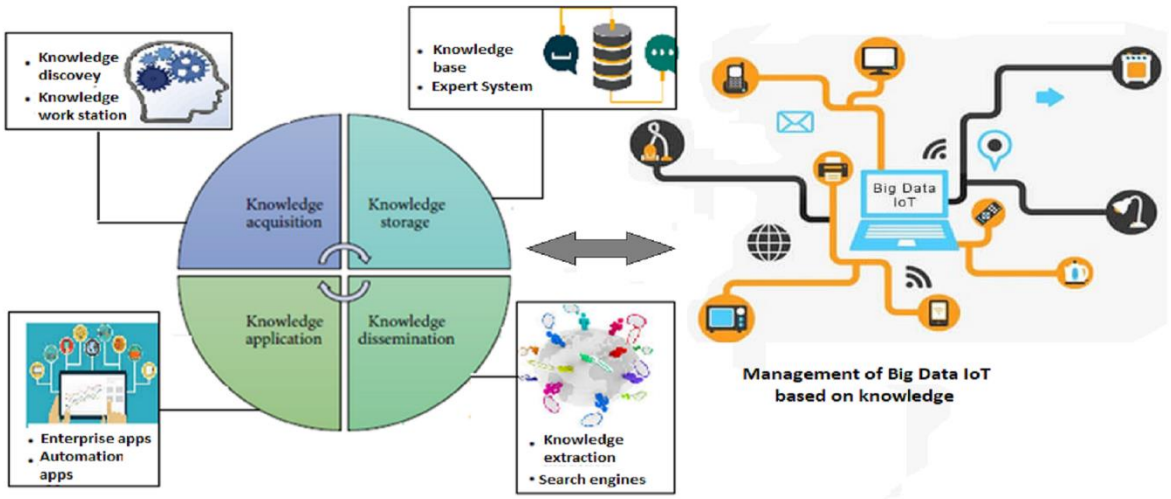


Figura 5. Sistema de Exploración del Conocimiento IoT. Adaptado de [16].

Adquisición de conocimiento se descubre a través del uso de varias técnicas de inteligencia tradicionales y computacionales. Las bases de datos de conocimiento almacenan la información importante descubierta y los sistemas expertos generalmente se diseñan en función del conocimiento descubierto. La difusión del conocimiento es importante para obtener información significativa de la base de conocimiento. La extracción de conocimiento es un proceso que busca información importante dentro de documentos, bases de conocimiento. La aplicación del conocimiento se emplea en varias aplicaciones [38].

2.4.2. Aplicaciones de Minería de *Big Data* en IoT

Las directrices de esta investigación se enfocan en un caso de uso para *Big Data* específico de aplicaciones IoT en el campo de "Cadenas de Suministro Inteligentes". En la Tabla 3 ,se muestra una comparación de las aplicaciones de la minería de *Big Data* para IOT.

Aplicaciones	**Método**				
	Clasificación	**Clustering**	**Reglas de Asociación**	**Predicción**	**Series de Tiempo**
Gobierno Electrónico	x	x	x		x
Análisis de Redes Sociales		x		x	x
Procesamiento del Lenguaje Natural (PNL)	x				
Bioinformática		x	x		
Reconocimiento de Voz	x				x
Industria	x	x	x		
Análisis de mercado		x	x	x	
Genética Humana		x			
Imágenes médicas	x	x			x
Cuidado de la salud		x	x		
Gestión de desastres				x	x

Tabla 3.Aplicaciones de minería de *Big Data* para IoT

2.5. Small Data

Small Data describe lo que los dispositivos están haciendo en el momento y en el tiempo. Al considerar las necesidades del análisis de IoT en un proyecto, la mayoría de los proyectos IoT que están basados en sensores van a confiar más en *Small Data* que *Big Data* [39].

2.5.1. *Small Data* en el futuro de IoT

Dispositivos de *Big Data* recogen cantidades masivas de información en bruto y rápidamente se convertirán en cuellos de botella debido al limitado Ancho de Banda de Internet y aumento de Latencia [40].

Small Data desencadena eventos basados en lo que está sucediendo ahora. Esos eventos se pueden combinar con información de comportamiento o de tendencias derivadas de algoritmos de *Machine Learning* que se ejecutan contra grandes conjuntos de datos [41].

Small Data a diferencia de *Big Data*, es procesado y manipulado antes de la transmisión, en lugar de enviar un gran flujo de datos sin procesar, la información se analiza primero para producir información pequeña y útil antes de que se transmita a través de Internet, ayudando a reducir la cantidad de ancho de banda necesaria para la transferencia. Su procesamiento se basa en *Fog Computing* en lugar de *Cloud Computing* [40]. La Figura 6 ,muestra los 4 principios clave del uso de *Small Data* .

Figura 6. Los 4 principios clave de *Small Data*.

Fog Computing es una arquitectura que depende de varios dispositivos, en lugar de un concentrador de datos basado en la nube; analiza los datos antes de la transmisión. Futuros dispositivos IoT se evaluarán en función de su capacidad de ofrecer resultados en tiempo real. *Small Data* permite una transferencia de información más eficiente y eficaz [40].

2.6. Comparación de Small Data vs. Big Data

Small Data conecta a las personas con información oportuna y significativa (derivada de *Big Data*), organizada y empaquetada para ser accesible, comprensible y accionable para las

tareas cotidianas. Esta definición se aplica a los datos que tenemos, así como a las aplicaciones de usuario final [42].
Big Data no es un requisito para todos los casos de uso de IOT. *Small Data* sabe lo que está haciendo un objeto rastreado. Si se requiere saber porque el objeto está haciendo eso, entonces se recurre a *Big Data* [41]. La Tabla 4 ,muestra una comparación resumida de las principales características entre *Big Data* y *Small Data*.

Categoría	*Big Data*	*Small Data*
Fuentes de Datos	Datos generados fuera de la empresa de fuentes de datos no tradicionales: • Social Media • Datos de Sensores • Datos de Logs • Datos de dispositivos • Video , imágenes, etc.	Datos empresariales tradicionales. Incluye : • Datos Transaccionales de ERP (*Enterprise Resource Planning*) • Datos de Sensores • Sistemas CRM (*Customer Relationship Managemen*t) • Transacciones Web • Datos Financieros
Volumen	• Terabytes 10^{12} • Petabytes 10^{15} • Exabytes 10^{18} • Zettabytes 10^{21}	• Gigabytes 10^{9} • Terabytes 10^{12}
Velocidad	• A menudo en tiempo real • Requiere respuesta inmediata	• *Batch* o casi en tiempo-real • No siempre requiere respuesta inmediata
Variedad	• Datos Estructurados • Datos No estructurados • Datos Multi-estructurados	• Datos Estructurados • Datos No estructurados • Inteligencia de Negocios, análisis e informes
Valor	• Análisis de negocios complejos, avanzados y predictivos	• Inteligencia de Negocios, análisis e informes
Costo	• Costoso	• Económico

Tabla 4. Comparación de *Small Data vs. Big Data*

2.7. Cadenas de Suministro Inteligentes

A continuación, se realiza una breve introducción a los conceptos clave dentro de este campo.

2.7.1. Cadena de Suministro

Red integrada de recursos y procesos que es responsable de la adquisición de materias primas, la transformación de estas materiales en productos intermedios y terminados, y la distribución de los productos terminados a los clientes finales [43].

Una cadena de suministro es un grupo de empresas que mueven los productos a la siguiente empresa. Por lo general, un conjunto de empresas que actúan por su cuenta se incluye en la producción y distribución de un bien para el cliente final [43].

2.7.2. Gestión de la Cadena de Suministro

Se define como la estrategia sistémica y estratégica de coordinación de las funciones comerciales tradicionales y las tácticas estas funciones comerciales dentro de una empresa

en particular y en todos los negocios dentro de la cadena de suministro, con el fin de mejorar el largo plazo el desempeño de las empresas individuales y la cadena de suministro en general [44].

2.7.3. BDA en IoT para Cadenas de Suministro

IoT y *Machine Learning* son utilizadas en el mantenimiento predictivo del activo para evitar paradas no planificadas. IoT puede proporcionar datos de telemetría en tiempo real para revelar los detalles de los procesos de producción y entrega [3].

El análisis de *Big Data* en IoT permite a una cadena de suministro ejecutar decisiones y controlar el entorno externo. La visibilidad en tránsito es otro caso de uso que desempeñará un papel vital en futuras cadenas de suministro en presencia de infraestructura IoT [1].

Los datos recolectados a través de las tecnologías RFID y GPS permitirán a los administradores de la cadena de suministro mejorar el envío automatizado, control de inventario en tiempo real y la información precisa de la entrega al predecir el tiempo de llegada [45]. La Figura 7, muestra el modelo de Negocio que utilizan las cadenas de suministro.

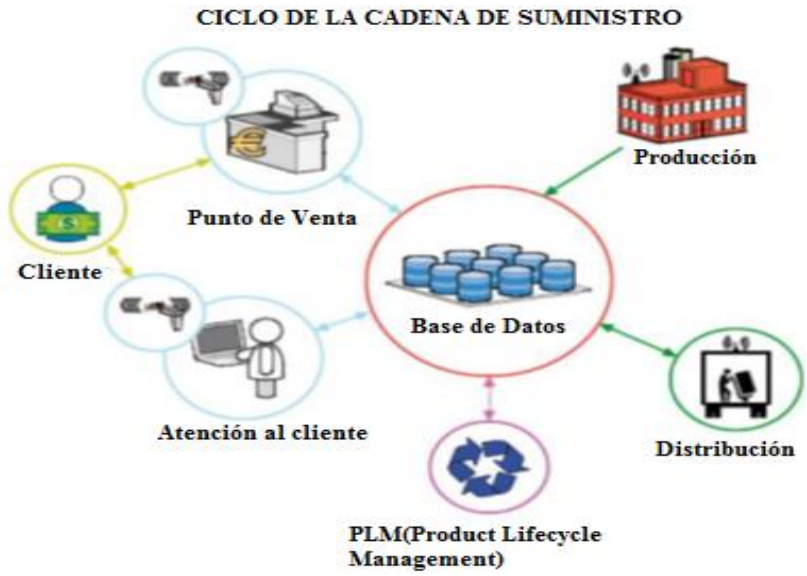


Figura 7. Modelo de negocio de las Cadenas de Suministro.

2.7.4. Aplicaciones de *Big Data* en operaciones de la Cadena de Suministro

Las soluciones de *Big Data* aplicadas a toda la cadena de suministro pueden implicar altos costos, haciendo que los responsables de la cadena de suministro sean más selectivos en la personalización de soluciones para operaciones específicas [3]. La Figura 8, muestra las operaciones en todo el ciclo de la cadena de suministro de extremo a extremo y la Tabla 5, describe a detalle cada una de sus operaciones .

Figura 8. Operaciones de la Cadena de Suministro. Adaptado de [3].

Entrega	Seguimiento de rutas de entrega, datos de tráfico, clima en tiempo real y re encaminamiento si es necesario para capacidad y compartir activos
Programación	Mayor visibilidad de los niveles de inventario, demanda y capacidad de fabricación
Almacenamiento	Realización en tiempo real de un gran análisis de datos dentro del sistema ERP del almacén e identificación de los niveles de inventario, emparejamientos de entrega y entregas
Previsión de la demanda	Estimación más precisa de la demanda mediante el acceso a los datos de ventas, las tendencias del mercado, datos de competidores y factores económicos locales y globales relevantes.
Distribución	Optimización en tiempo real de complejas redes de centros de distribución y almacenes basados en los datos de flujo de materiales
Planificación del inventario	Transparencia total en el nivel de SKU (*Stock-keeping unit*) y sistemas de reposición totalmente automatizados combinados con datos de pronóstico de demanda que eliminan bajo / *overstocking*(exceso de stock).

Tabla 5. Operaciones en la Cadena de Suministro.

2.7.5. Capacidades necesarias para BDA en la Cadena de Suministro

El poder de *Big Data* y las oportunidades que ofrece para mejorar las cadenas de suministro se han hecho imprescindibles. A medida que la complejidad aumenta, la capacidad de analizar y obtener ideas significativas y oportunas se convertirá en el centro de las organizaciones [3]. La Tabla 6, muestra algunas de las principales capacidades necesarias para los analistas de datos de la cadena de suministro.

Aplicación de Capacidad		Descripción de la capacidad
	Estadísticas de las Cadena de Suministro	Conciencia de los métodos de estimación estadística y muestreo
	Previsión de la Cadena de Suministro	Comprensión de los métodos cualitativos y cuantitativos de previsión
	Optimización de la Cadena de Suministro	Capacidad para adoptar métodos analíticos y numéricos de optimización
	Simulación de la Cadena de Suministro	Rediseño de los procesos de la cadena de suministro utilizando modelos de simulación, visualización de datos y repositorios de datos

Tabla 6. Principales capacidades necesarias para BDA en la Cadena de Suministro

2.7.6. Beneficios de IoT en Cadenas de Suministro

Al incorporar esta tecnología en la cadena de suministro se obtienen beneficios como:

- La visibilidad del estado de un producto en tiempo real;
- Mejora en la productividad;
- Mejor alineación entre la planeación y la ejecución;
- Mejora de la eficiencia y efectividad del transporte y la logística;
- Habilitación de la colaboración en la cadena de suministro.

La utilización de *Big Data* en las cadenas de suministro, ha llegado la evolución que conduce a la transformación de los procesos de negocio. Ejemplos de este tipo de utilización son los sistemas de pedidos automáticos [46].

2.7.7. Identificación de fuentes de *Big Data* en Cadenas de Suministro

Big Data de RFID y GPS pueden ayudar en el posicionamiento de inventario en tiempo real y almacenamiento. El punto de venta (POS) es uno de los principales factores que facilitan la predicción de la demanda y el análisis del comportamiento del cliente [3].

Las cadenas de suministro del futuro estarán impulsadas por sofisticados algoritmos, simulaciones y análisis prescriptivo que permitirán la toma de decisiones en toda la empresa. Este nivel de BDA ya está permitiendo que las organizaciones comprendan realmente el costo de servir y la contribución económica en un nivel granular a través de producto, proveedor, distribuidor, cliente, unidad de negocio y geografía. Pueden ayudar a encontrar un mejor equilibrio entre la oferta y la demanda [3]. Las 5 principales fuentes de *Big Data* en las Cadenas de Suministro son presentadas en la Figura 9.

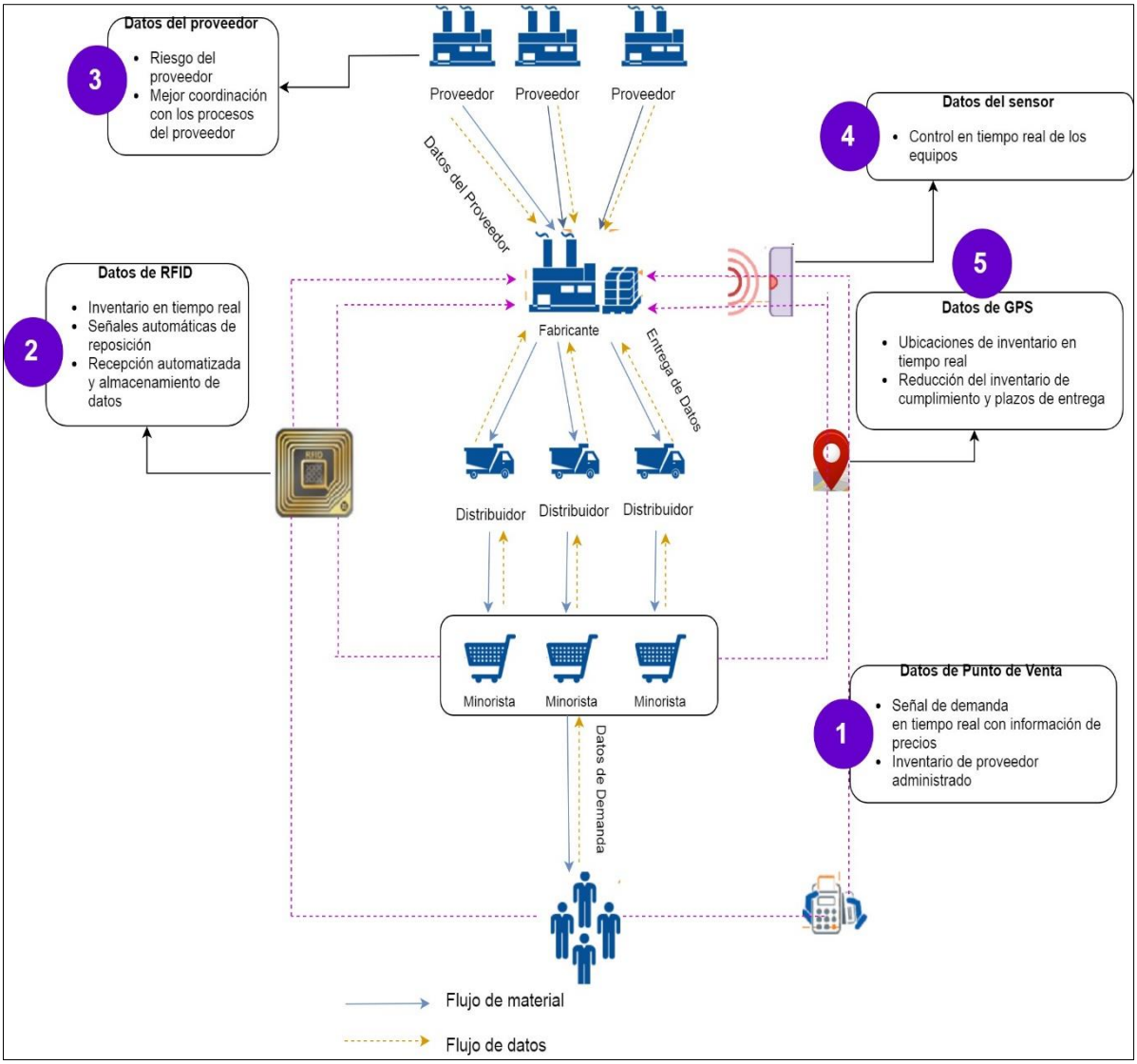


Figura 9. Fuentes de *Big Data* en Cadenas de Suministro futuras. Adaptado de [3].

2.8. Investigación de Trabajos Relacionados

La siguiente sección presenta investigaciones y trabajos relacionados con *Big Data y Small Data* en IoT.

2.8.1. IoT + *Small Data*: en análisis y servicios de una tienda minorista

Un enfoque de *Small Data* basada en datos analíticos de una tienda minorista, la combinación de datos de sensores, dispositivos portátiles personales y sensores desplegados en tienda y dispositivos IoT, se utiliza para crear servicios individualizados [47]. Entre los desafíos clave se incluyen: Minería de sensores y Desencadenamiento juicioso.

La Minería de datos de sensores para capturar el nivel de productividad de un comprador y sus interacciones. El Desencadenamiento juicioso (Ejemplo, una cámara) para capturar sólo

partes relevantes de las actividades de un comprador en la tienda. Se llevó a cabo experimentos con 5 *smartwatch* de usuarios que interactuaron con los objetos colocados en un laboratorio (imitando las interacciones correspondientes del supermercado) [47].

Los resultados iniciales muestran: 94% de precisión en identificar un gesto de selección de artículos, 85% de precisión en la identificación la ubicación de la estantería desde donde se recogió el artículo y el 61% precisión en la identificación del elemento exacto recogido (a través del análisis de los datos de la cámara del smartwatch) [47].

Este tipo de "Small Data Analytics" (información detallada sobre un comprador individual) puede (a) inferir las acciones en tienda de un comprador y las opciones del producto en tiempo real (incluso antes del contador de pago), y (b) es barato y fácil de implementar (no requiere infraestructura compleja apoyo). El seguimiento en su smartphone, y (c) recomendaciones basadas únicamente en perfiles de clientes generales y de largo plazo [47]. La Figura 10 ilustra la arquitectura del sistema.

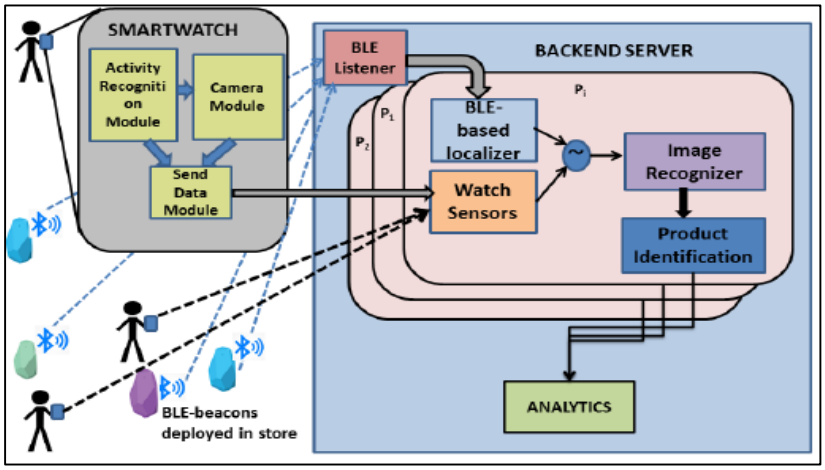


Figura 10. Arquitectura del Sistema [47].

2.8.2. Desarrollo de una *Smart City* utilizando IoT y *Big Data*

El rápido crecimiento de la densidad de población en las zonas urbanas requiere un medio, la ciudad inteligente es una tarea difícil. Por lo tanto, se ha propuesto un sistema para el desarrollo inteligente de la ciudad basado en IoT utilizando BDA. Se utilizó el despliegue de sensores, incluyendo sensores inteligentes para el hogar, redes de vehículos, sensores meteorológicos y de agua, sensores de aparcamiento inteligentes y objetos de vigilancia [48]. Inicialmente se propone una arquitectura de cuatro niveles que incluye:

1. **Nivel inferior:** Responsable de recursos IOT, generaciones de datos y colecciones.
2. **Nivel intermedio 1:** Gestión de todos los tipos de comunicación entre sensores, relés, estaciones base, Internet.
3. **Nivel intermedio 2:** Administración, procesamiento de datos utilizando el marco Hadoop.

4. **Nivel superior:** Responsable de la aplicación, uso del análisis de datos y resultados generados. Los datos recolectados de todo el sistema inteligente se procesan en tiempo real para lograr ciudades inteligentes utilizando Hadoop con Spark, VoltDB o Storm [48].

2.8.3. Hacia un *framework* IoT BDA (IBDA): Sistemas de edificios inteligentes

Existe un creciente interés por los edificios inteligentes habilitados para IoT. Sin embargo, el almacenamiento y el análisis de gran cantidad de datos de alta velocidad en tiempo real es una tarea difícil. Existe la necesidad de un marco integrado de análisis de datos grandes (IBDA). Este trabajo presenta un marco IBDA para el almacenamiento y análisis de datos en tiempo real generados a partir de sensores IoT dentro del edificio inteligente [49].

La versión inicial del marco de IBDA se ha desarrollado utilizando Python y la plataforma *Big Data Cloudera*. El framework demuestra con la ayuda de un escenario que implica el análisis de datos de edificios inteligentes en tiempo real para gestionar automáticamente el nivel de oxígeno, luminosidad y gases peligrosos en diferentes partes del edificio [49]. La clave de este trabajo es la integración de BDA y IoT, como se muestra en la Figura 11.

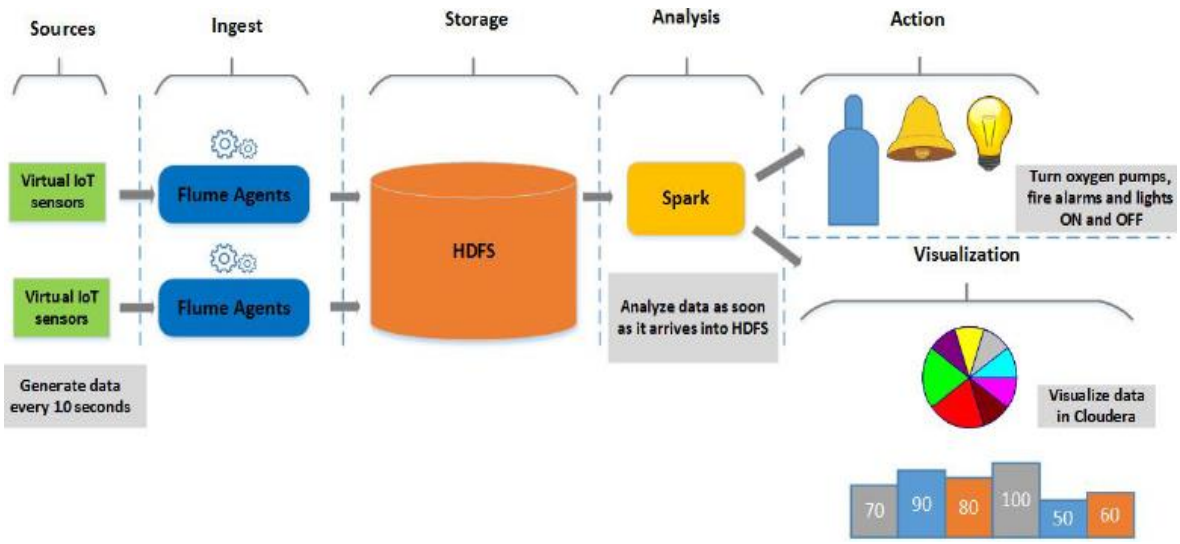


Figura 11. Arquitectura IBDA - ingerir, almacenar, analizar y actuar [49].

Esta investigación es explicada en tres pasos principales: Revisión de Literatura, Diseño y Evaluación. La Revisión de Literatura - relacionada con el marco de análisis de datos en tiempo real para los datos generados por IOT y las técnicas para controlar los edificios inteligentes en tiempo real. El Diseño del marco de IBDA para análisis de datos en tiempo real para edificios inteligentes utiliza sensores IoT y Apache Hadoop. La Evaluación utiliza un problema del mundo real para un escenario de construcción inteligente. El alcance de este marco se limita a la generación de datos, extracción de datos, ingesta de datos en HDFS (Hadoop *Distributed File System*), visualización de datos, análisis de datos y control en tiempo real del edificio inteligente [49].

2.8.4. Paradigmas de compresión *de Big Data* para soportar *frameworks* de IoT intensivos en datos eficaces y escalables

Se centra en los grandes paradigmas de compresión de datos dentro de los marcos de referencia de IoT basados en datos de referencia, abarca infraestructuras orientadas al servicio, Cloud Computing, gestión de datos y análisis [50].

Básicamente, las grandes técnicas de compresión de datos le permiten dominar la complejidad de las grandes tareas de gestión de datos dentro de estos marcos, influyendo así en una forma beneficiosa todas las otras actividades, tal vez entregadas como servicios en una arquitectura de referencia de Cloud [50].

En esta investigación se ofrece una visión general de las técnicas de compresión de datos más avanzadas y grandes, se describen las direcciones futuras de la investigación sobre el tema científico bajo investigación que se considerará en los próximos años [50].

2.9. Síntesis

En este capítulo se discutió la relación entre BDA e IoT, se examinaron varios temas de investigación, varias oportunidades generadas por el análisis de datos en el paradigma de IoT, desafíos y herramientas utilizadas para BDA.

La interacción entre IoT y *Big Data* está actualmente en una etapa donde se procesa, transforma y analiza grandes cantidades de datos a una alta frecuencia necesaria. *Big Data* permite tomar mejores decisiones, pero a medida que la confianza aumenta en los algoritmos, los datos y los análisis, la cuestión de confianza emerge como una consideración importante. Las diferentes técnicas utilizadas para el análisis incluyen Análisis Estadístico, Aprendizaje Automático, Extracción de Datos, Análisis Inteligente, Computación en la Nube y Procesamiento de Flujo de Datos.

El uso correcto de *Big Data* contiene la clave para mejorar la madurez de la Cadena de Suministro asegurando la integridad de los datos, una mayor visibilidad y control, aumentando la agilidad y la capacidad de respuesta.

Las oportunidades que brinda *Big Data* para la mejora en las Cadenas de Suministro se han vuelto imperativas a medida que la complejidad de estas aumenta, la capacidad de analizar y obtener conocimientos significativos y oportunos se vuelve fundamental para las organizaciones. Las organizaciones que no lo hacen se vuelven irrelevantes rápidamente.

3. Herramientas para procesamiento de *Big Data*

Uno de los objetivos del uso de las tecnologías *Big Data es* el de transformar los datos en conocimiento útil para la empresa, para lo cual las herramientas de BDA, ayudan a procesar y almacenar todos los datos recolectados. En esta sección, se discuten algunas técnicas actuales para su almacenamiento y análisis y con un enfoque en tres herramientas emergentes importantes, como MapReduce [19], Apache Spark [51] y Storm [52].

La mayoría de las herramientas disponibles se centran en el procesamiento por lotes, el procesamiento de flujo y el análisis interactivo. La mayoría de las herramientas de procesamiento por lotes se basan en la infraestructura de Apache Hadoop [53], como Mahout [54] y Dryad [55]. Las aplicaciones de datos de flujo se utilizan principalmente para el análisis en tiempo real. Un ejemplo de plataforma de transmisión a gran escala es Splunk [56]. Dremel y Apache Drill [57] son plataformas de *Big Data* que respaldan el análisis interactivo. Estas herramientas son muy útiles para el desarrollo de proyectos *Big Data*. El flujo de trabajo típico de los proyectos *Big Data* es discutido por Huang et [58]. La Figura 12, presenta el flujo de trabajo de un proyecto *Big Data*.

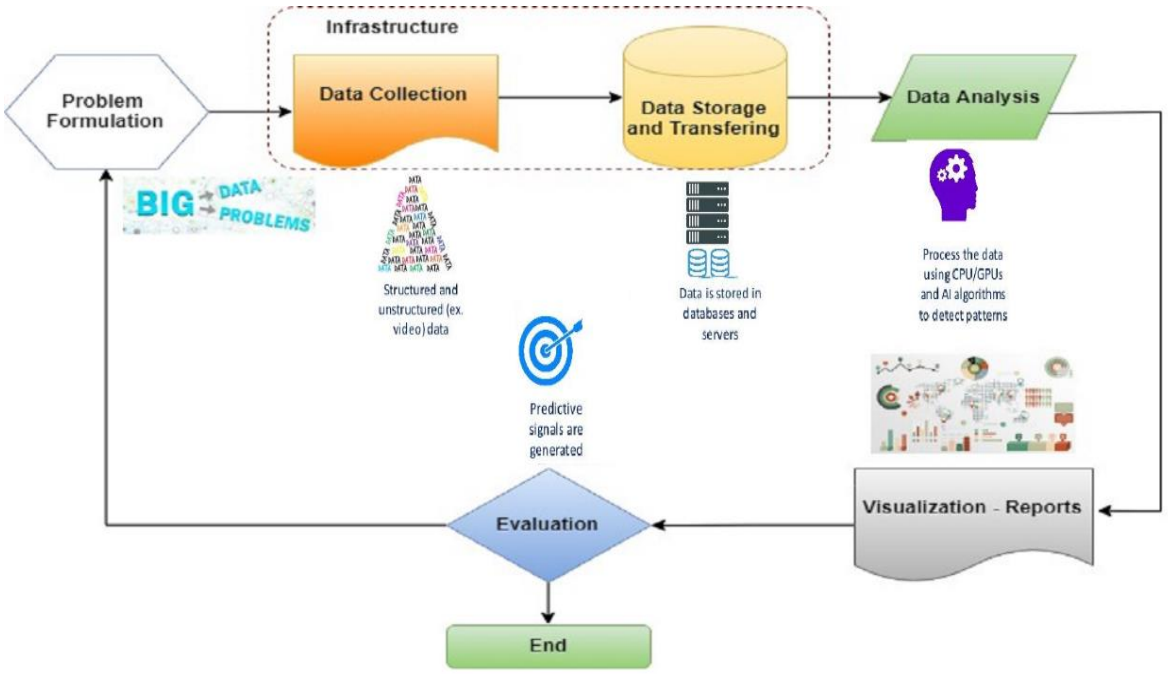


Figura 12.Flujo de trabajo de un proyecto *Big Data*. Adaptado de [58].

3.1. Apache Hadoop

Esta herramienta de *Big Data open source* se considera el *framework* estándar para el almacenamiento de grandes volúmenes de datos; se usa también para analizar y procesar, y es utilizado por empresas como Facebook y Yahoo [18].

La biblioteca Hadoop utiliza modelos de programación simples para el almacenamiento y procesamiento distribuido de grandes conjuntos de datos en clústeres, dando redundancia y al mismo tiempo, aprovechando muchos procesos a la vez. Dispone de un sistema de archivos distribuido en cada nodo del clúster: el HDFS y se basa en el proceso MapReduce de dos fases [53].

La combinación de estos dos permite que los datos estén replicados y distribuidos por N nodos beneficiando la capacidad de acceso a grandes volúmenes. Cuando se ejecuta alguna operación sobre datos distribuidos, Hadoop se encarga de procesar cada porción de los datos en el nodo que los contiene, permite escalar de forma casi lineal. Del almacenamiento se encarga HDFS y del procesamiento MapReduce [53]. Hadoop puede ejecutarse de 3 formas diferentes, de acuerdo a la distribución de procesos:

1. ***Standalone mode***: Modo predeterminado provisto con Hadoop. Todo se ejecuta como un solo proceso.
2. ***Pseudo-distributed mode***: Hadoop está configurado para ejecutarse en una sola máquina, con diferentes daemons de Hadoop ejecutados como diferentes procesos de Java.
3. ***Fully distributed* o *cluster mode:*** Aquí, una máquina en el clúster normalmente se etiqueta como NameNode y otra máquina se designa como JobTracker. Solo se coloca un NameNode en cada clúster, que administra el espacio de nombres, los metadatos del sistema de archivos y el control de acceso [59].

3.1.1. Arquitectura principal de Hadoop

A continuación, se presentan los componentes principales de la arquitectura de Hadoop.

3.1.1.1. Hadoop Distributed File System (HDFS)

HDFS optimiza grandes flujos y trabaja con ficheros grandes en sus lecturas y escrituras. Su diseño reduce la entrada/salida en la red. La escalabilidad y disponibilidad son otras de sus claves, gracias a la replicación de los datos y tolerancia a los fallos [53]. La Figura 13, muestra un esquema de almacenamiento , lectura y escritura en HDFS.

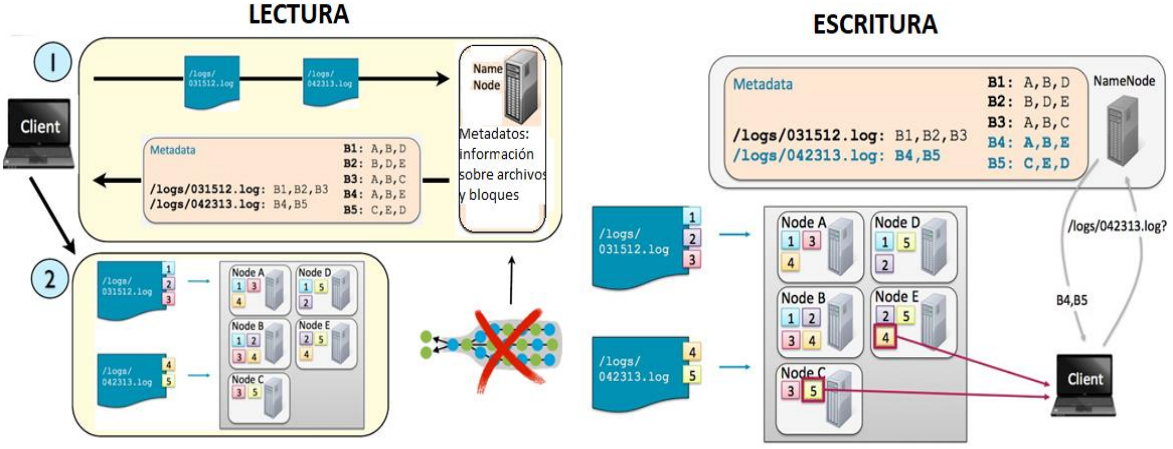


Figura 13. Procesos de lectura y escritura en HDFS.

Su arquitectura incluye tres tipos de nodos Master, Worker y Client. *Master* supervisa las operaciones de almacenamiento de datos en HDFS y la ejecución de cómputos paralelos en esos datos utilizando MapReduce. Los Workers realizan el trabajo de almacenar los datos y ejecutar los cálculos. *Client* carga datos en el clúster, envía trabajos de MapReduce que describen cómo deben procesarse esos datos y luego recupera o visualiza los resultados del trabajo cuando finaliza el procesamiento [59].

Un archivo puede dividirse en uno o más bloques de datos, y estos bloques de datos se mantienen en un conjunto de *DataNodes* [59]. A continuación, se describen los elementos importantes del clúster:

- ***NameNode**:* Se encuentra solo uno en el clúster. Regula el acceso a los ficheros por parte de los clientes. Mantiene en memoria la metadata del sistema de ficheros y control de los bloques de fichero que tiene cada *DataNode* [59].
- ***DataNode**:* Lee y escribe las peticiones de los clientes. Los ficheros están formados por bloques, estos se encuentran replicados en diferentes nodos [59].

3.1.1.2. MapReduce

Es un modelo de programación para el procesamiento de grandes conjuntos de datos. Se basa en el método de dividir y conquistar. Simplifica el procesamiento en paralelo separando la complejidad que existen en los sistemas distribuidos [16].

Las funciones *Map* transforman un conjunto de datos a un número de pares ***key/value***. La función *Reduce* es usada para combinar los valores (con la misma clave) en un mismo resultado [53].

- ***Job Tracker*:** Es un *Daemon* esencial para la ejecución de MapReduce, recibe las solicitudes para la ejecución de MapReduce desde el cliente. Coordina el sistema de procesamiento de datos para Hadoop. El proceso de *JobTracker* se ejecuta en un nodo separado y no en un *DataNode* [60].
- ***TaskTracker*:** Es un nodo en el clúster que acepta tareas: Operaciones de *Map*, *Reduce* y *Shuffle* de un JobTracker. Se ejecuta en *DataNode*. Las tareas *Mappe*r y *Reducer* se ejecutan en *DataNodes* administrados por *TaskTrackers* [60]. Cuando un T*askTracker* no responde, *JobTracker* asignará la tarea ejecutada por el *TaskTracker* a otro nodo [60].

La Figura 7, muestra la descripción de *Mappers, Reducers, Partitiones y Combiners.*

Mappers	Requerido para generar un número arbitrario de pares intermedios.
Reducers	Se aplica a todos los valores intermedios asociados con la misma clave intermedia.
Partitioners	Su trabajo principal es dividir el espacio clave intermedio, y luego asignar los pares intermedios clave-valor intermedio a reductores.
Combiners	Los *Combiners* son una optimización (opcional).
	Antes de realizar la fase de *shuffle* and *sort*, permite la agregación local de datos.
	Se utilizan para ahorrar ancho de banda, por ejemplo, el programa de recuento de palabras.

Tabla 7. Descripción de *Mappers, Reducers, Partitiones y Combiners.*

A continuación se presenta en la Figura 14, la arquitectura de alto nivel de Hadoop.

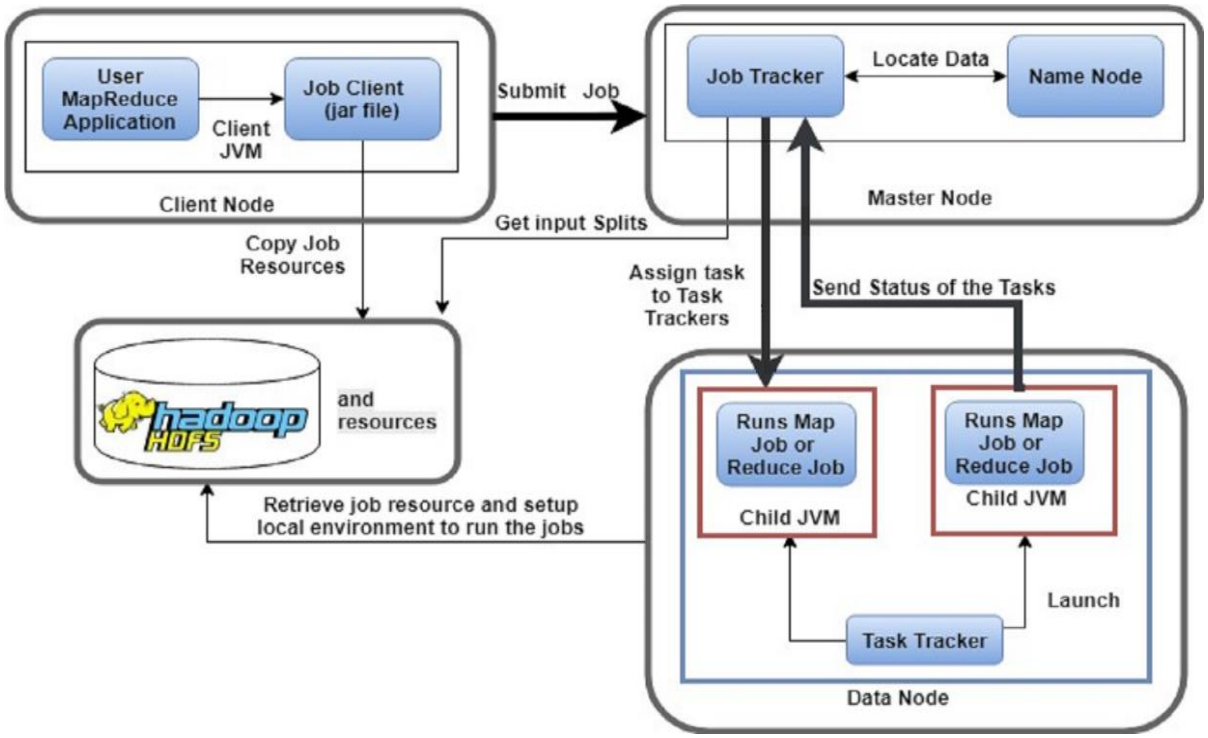


Figura 14. Arquitectura de alto nivel de Hadoop. Arquitectura de alto nivel de Hadoop. Adaptado de [16].

Los programas de MapReduce son generalmente escritos en Java. También se pueden codificar en otros idiomas, como C ++, Python, Ruby, R. Estos programas pueden procesar datos almacenados en diferentes archivos y sistemas de bases de datos. En Google, por ejemplo, MapReduce se implementó sobre Google File System (GFS) [59].

3.1.2. Ecosistema de Hadoop

En Hadoop se tiene un ecosistema muy diverso, que crece día tras día. A continuación, la Figura 15 , muestra los componentes principales del ecosistema de Hadoop [53].

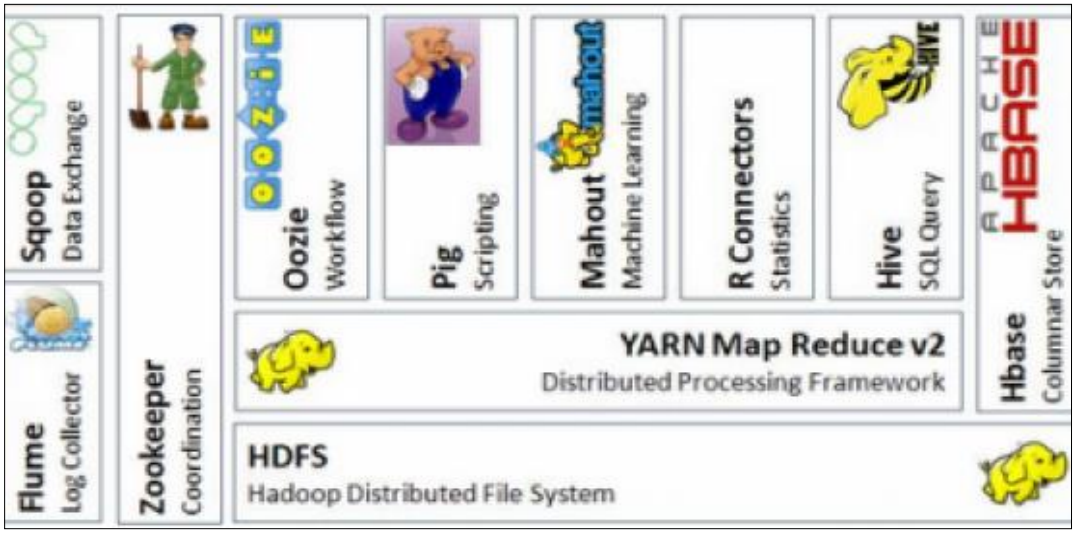


Figura 15. Ecosistema de Hadoop [53].

3.1.2.1. Apache Flume

Sistema Distribuido de captura, agregación y movimiento de grandes cantidades de datos, *logs* de diferentes servidores a un repositorio central, simplificando el proceso de recolectar estos datos para almacenarlos en Hadoop y poder analizarlos [53]. La Figura 16, presenta el proceso de trabajo de Apache Flume.

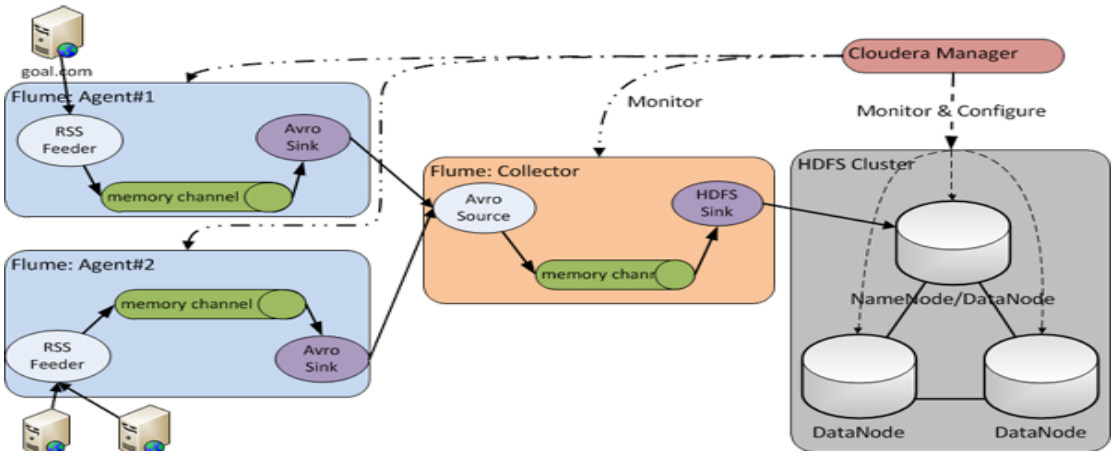


Figura 16. Sistema Distribuido Apache Flume [53].

3.1.2.2. Apache Hive

Es un Sistema de DW para Hadoop facilitando el uso de agregación de 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 [53]. La Figura 17, muestra la integración de Hadoop con Hive.

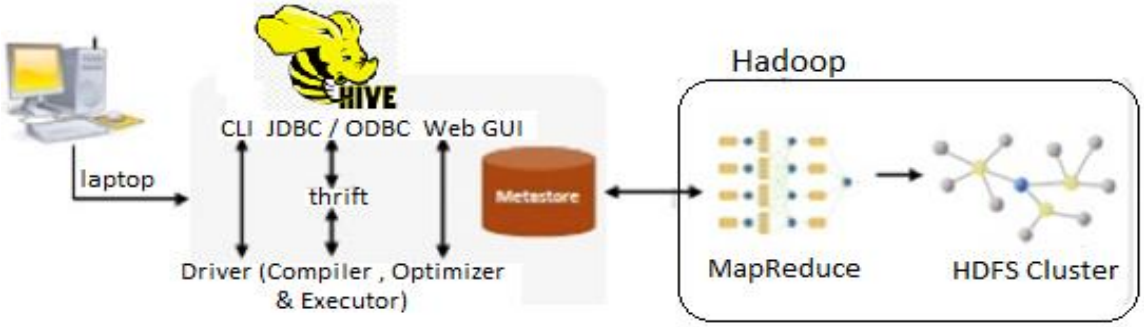


Figura 17.Sistema Hive. Adaptado de [53].

3.1.2.3. Apache HBase

Conocida como base de datos de Hadoop. Es un 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, no sigue el esquema relacional. No admite SQL [53]. La Figura 18, indica su esquema de funcionamiento.

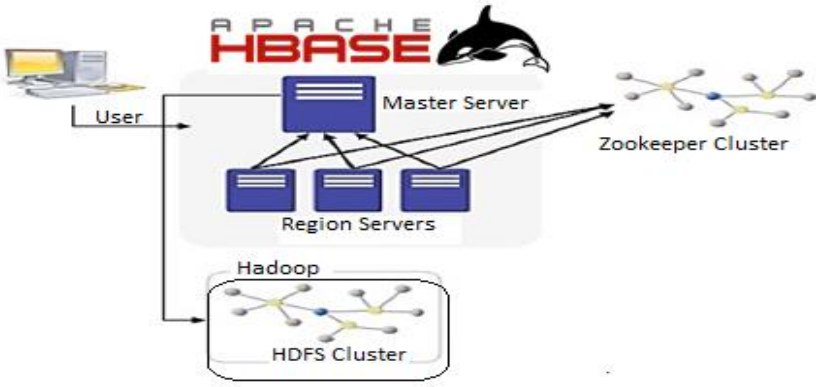


Figura 18. Base de Datos Apache HBase. Adaptado de [53].

3.1.2.4. Apache Mahout

Su objetivo es proporcionar técnicas de *Machine Learning* para aplicaciones de análisis de datos a gran escala e inteligentes. Los algoritmos básicos de Mahout, incluyendo Agrupación, Clasificación, Extracción de patrones, Regresión, Reducción de dimensiones, Algoritmos Evolutivos y filtrado colaborativo basado en lotes. Se ejecutan en la plataforma Hadoop a través del *framework* MapReduce. Empresas como Google, IBM, Amazon, Yahoo, Twitter y Facebook han implementado algoritmos de aprendizaje escalable de máquina [54]. La Figura 19 muestra un esquema básico de funcionamiento.

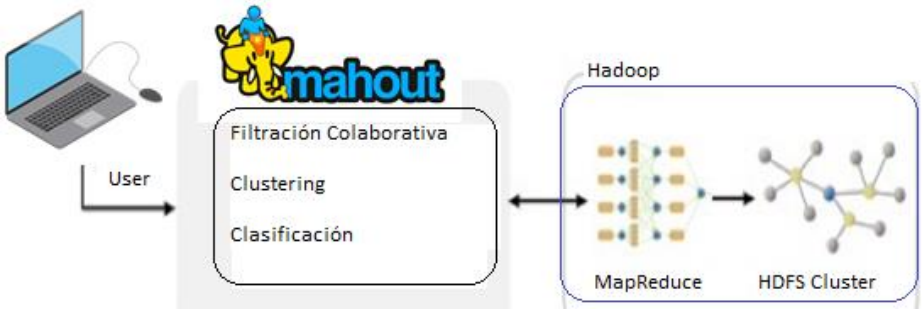


Figura 19. Apache Mahout. Adaptado de [53].

3.1.2.5. Apache Sqoop

Sqoop, "Sql-to-Hadoop", es una herramienta diseñada para transferencia de datos voluminosos entre Hadoop y sistemas de almacenamiento con datos estructurados, como bases de datos relacionales [53]. La Figura 20, muestra su arquitectura funcional.

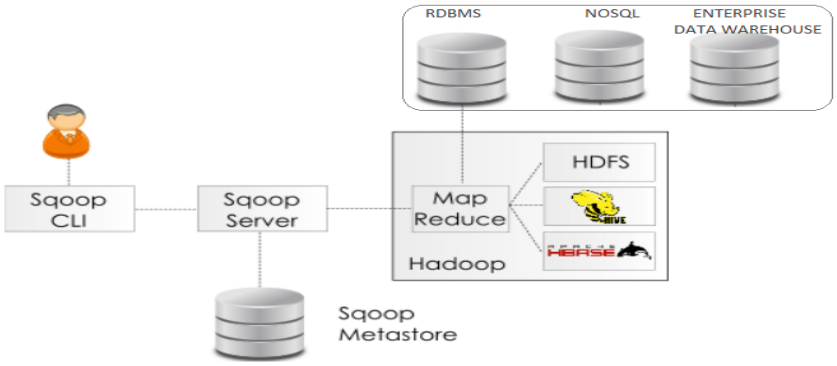


Figura 20. Apache Sqoop. Adaptado de [53].

Algunas de las características de Apache Sqoop son:

- Importación de tablas individuales o bases de datos enteras a HDFS;
- Generación de clases Java que permiten interactuar con los datos importados;
- Importación de bases de datos SQL a Hive [53].

3.1.2.6. Apache Pig

Desarrollado por Yahoo, permite a los usuarios de Hadoop centrarse más en el análisis de los datos y menos en la creación de programas MapReduce. Para simplificar el análisis proporciona un lenguaje procedural de alto nivel, para trabajar en cualquier tipo de datos [53].La Figura 21, muestra la arquitectura funcional de Apache Pig. Consta de dos componentes:

- El lenguaje PigLatin;
- El entorno de ejecución, donde los programas Pig Latin se ejecutan [53].

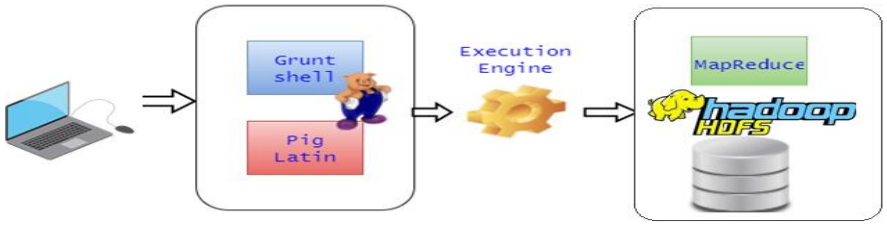


Figura 21. Apache Pig

3.1.2.7. Apache Impala

Impala proporciona consultas SQL rápidas e interactivas directamente en sus datos de Apache Hadoop almacenados en HDFS, HBase o *Amazon Simple Storage Service* (S3) [61]. Utiliza los mismos metadatos de la sintaxis SQL (Hive SQL), el controlador ODBC y la interfaz de usuario Hue. Esto proporciona una plataforma familiar y unificada para consultas en tiempo real o por lotes[61]. Sus características son:

- Interfaz SQL para consulta de grandes volúmenes de datos en Apache Hadoop;
- Consultas distribuidas en un entorno de clúster;
- Posibilidad de compartir archivos de datos entre diferentes componentes sin copia ni paso de exportación / importación; por ejemplo, para escribir con Pig, transformar con Hive y consultar con Impala [61].

3.1.2.8. Apache Solr

Apache Solr es un motor de búsqueda de texto de código abierto que permite indexar documentos, haciendo más fácil la búsqueda usando consultas de forma libre de manera similar a Google. Solr organiza los datos de forma similar a una base de datos SQL. Cada registro se denomina *'document'* y consta de campos definidos por el esquema: como una fila en una tabla de base de datos. En lugar de una tabla, Solr lo llama una *"collection"* de documentos. Puede ingresar consultas de texto que coincidan parcialmente con un campo, para búsqueda de páginas web y en paralelo [62].

3.2. Apache Spark

Es un poderoso motor de procesamiento de código abierto construido en torno a la velocidad, facilidad de uso y análisis sofisticado. Desarrollado en UC Berkeley AMP Lab, en respuesta a limitaciones en el marco de procesamiento MapReduce. Basado en el disco de dos etapas de Hadoop, manteniendo la escalabilidad lineal, la tolerancia a fallas de MapReduce y expandiendo las capacidades de procesamiento en cuatro áreas importantes: Análisis en memoria; Federación de datos; Análisis iterativo; y Análisis casi en tiempo real [51].

El análisis en memoria permite el acceso en memoria a los resultados intermedios en una canalización de procesamiento de varias etapas a través de su abstracción de conjunto de *Resilient Distributed Datasets* (RDD), aumentando el rendimiento hasta 100 veces más rápido que MapReduce. La federación de datos tiene un amplio conjunto de bibliotecas y APIs; que permiten a los desarrolladores crear flujos de trabajo analíticos de manera más eficiente, para acceder a cualquier fuente de datos, desde HDFS o almacenamiento de objetos, hasta bases de datos relacionales y NoSQL [51].

El análisis de algoritmos altamente iterativos se usa para el aprendizaje automático y el análisis de gráficos. Bibliotecas como GraphX de Spark unifican el análisis de gráficos iterativos con *Extract, Transform, Load* (ETL), y el análisis interactivo casi en tiempo real para la integración con herramientas en el ecosistema de Hadoop [51].

El "*Driver Program*" es el punto de inicio de la ejecución de una aplicación en Clúster de Spark. El "*Clúster Manager*" asigna recursos y los "*Worker Nodes*" para realizar el procesamiento de datos en forma de tareas. Cada aplicación tendrá un conjunto de procesos, llamados ejecutores. Su principal ventaja es el soporte para implementar aplicaciones Spark en un clúster Hadoop existente [63]. La Figura 22, muestra el diagrama de su arquitectura.

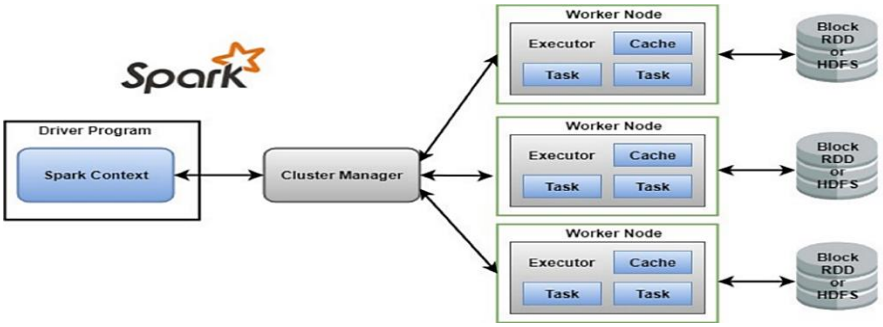


Figura 22. Diagrama de la arquitectura de Apache. Adaptado de [63] .

3.3. Dryad

Modelo de programación popular para implementar programas paralelos y distribuidos para manejar grandes bases de contexto en el gráfico de flujo de datos. Es un conjunto de nodos informáticos que permite utilizar los recursos de un clúster informático para ejecutar un programa de forma distribuida. Un usuario de Dryad usa miles de máquinas, cada una con múltiples procesadores o núcleos [55].

Su principal ventaja es que los usuarios no necesitan saber nada acerca de la programación simultánea. Una aplicación DRYAD ejecuta un gráfico computacional dirigido, que se compone de vértices computacionales y canales de comunicación. Por lo tanto, Dryad proporciona una gran cantidad de funcionalidades, incluida la generación de gráficos de tareas, programación de máquinas para procesos disponibles, manejo de fallas de transición en el clúster, recopilación de métricas de rendimiento y visualización del trabajo [55].

3.4. Storm

Es un sistema de computación en tiempo real distribuido y tolerante a fallas para procesar grandes datos de transmisión. Un Clúster Storm es similar al Clúster de Hadoop. En Storm, se ejecutan diferentes topologías para diferentes tareas de Storm, mientras que la plataforma Hadoop implementa trabajos de MapReduce para las aplicaciones correspondientes [52].

Un Storm Clúster se compone de dos tipos de nodos, como el *Master Node y Worker Node. Master Node y Worker Node* implementan dos tipos de funciones, como nimbo y supervisor, respectivamente. Los dos roles tienen funciones similares de acuerdo con Jobtracker y Tasktracker del *framework* MapReduce. Nimbus se encarga de distribuir el código a través del Storm Clúster, programar y asignar tareas a los *Worker Nodes* y supervisar todo el sistema [16]. El supervisor cumple las tareas asignadas por Nimbus [16].

3.5. Lenguaje de programación R

R permite el análisis y manejo estadístico, es uno de los más usados para la Minería de Datos, facilitando la manipulación de datos rápidamente. Paquetes como *plyr*[1], hacen que sea mucho más hábil y eficiente en la preparación de los datos para su posterior análisis. Aporta capacidades avanzadas, visualización de los datos y resultados e implementa una gran cantidad de algoritmos de *Machine Learning* [64].

Jan Wijffels menciono que el uso de R permite procesar fácilmente conjuntos de datos que de hasta un millón de registros. Conjuntos con aproximadamente millón a un billón de registros también se pueden procesar con un esfuerzo adicional. Conjuntos con más de mil millones de registros deben analizarse mediante algoritmos de reducción de mapas. Estos algoritmos pueden diseñarse en R y procesarse con conectores a Hadoop [64].

3.5.1. Estrategias de manejo de *Big Data* en R

Al analizar *Big Data* con R, se considera cinco estrategias diferentes: Muestreo , Hardware de Mayor Capacidad, Almacenamiento de Objetos en Disco Duro y Análisis a Nivel de Fragmento , Integración de Lenguajes con Mayor Rendimiento y Utilización Interpretes Alternativos [64].

[1] https://cran.r-project.org/web/packages/plyr/index.html

Muestreo: Permite un análisis en su totalidad y reduce su tamaño, disminuyendo significativamente el rendimiento de un modelo. Hadley Wickham menciona que la construcción de modelos basados en muestras es aceptable, al menos si el tamaño de los datos excede el umbral de mil millones de registros [64].

Hardware de Mayor Capacidad: R mantiene todos los objetos en memoria, por lo tanto requiere mayor capacidad; R puede abordar actualmente 8 TB de RAM si se ejecuta en máquinas de 64 bits y 2 GB de RAM direccionables en máquinas de 32 bits [64].

Almacenamiento de Objetos en Disco Duro y Análisis a Nivel de Fragmento: Existen paquetes disponibles que evitan almacenar datos en memoria almacenándolos en disco, como efecto secundario la fragmentación conduce a la paralelización, si los algoritmos permiten el análisis paralelo de los fragmentos en principio; su desventaja es que solo los algoritmos y las funciones R se pueden ejecutar en general [64].

Integración de Lenguajes de Programación de Mayor Rendimiento:Utiliza C ++ o Java para evitar cuellos de botella y el rendimiento de procedimientos costosos [64].

Utilización Interpretes Alternativos: Como pqR esbastante rápido , otro proyecto *open source* es Renjin [65], implementa el intérprete R en Java, se ejecuta en JVM (Java Virtual Machine); Oracle ofrece a Oracle R el uso gratuito que utiliza la biblioteca matemática Intel, por lo tanto, logra un mayor rendimiento sin cambiar el núcleo de R [64].

3.6. Apache Drill

Sistema distribuido para el análisis interactivo de *Big Data*. Tiene más flexibilidad para soportar muchos tipos de lenguajes de consulta, formatos de datos y fuentes de datos. También está especialmente diseñado para explotar datos anidados, su objetivo de ampliar a 10.000 servidores o más y alcanza la capacidad de procesar *petabytes* de datos y billones de registros en segundos. Drill usa HDFS para el almacenamiento y MapReduce para realizar análisis por lotes [57].

3.7. Splunk

Es una plataforma inteligente y en tiempo real desarrollada para la explotación de máquinas generadas por *Big Data*. Combina tecnologías en la nube y *Big Data*, permitiendo al usuario buscar, monitorear y analizar sus datos generados por la máquina a través de la interfaz web. Los resultados son presentados de forma intuitiva, como gráficos, informes y alertas. A diferencia de otras herramientas de procesamiento de flujo; ésta indexa datos estructurados y no estructurados generados por la máquina, búsqueda en tiempo real, informes de resultados analíticos y tableros. Proporciona diagnóstico de problemas para sistemas e infraestructuras de información, y soporte inteligente para operaciones comerciales [56].

3.8. Jaspersoft

Es un software de código abierto para análisis de datos escalables con capacidades de análisis en tiempo real, informes de columna de base de datos y tiene una capacidad de visualización

de datos rápida en plataformas de almacenamiento populares, como Mongo DB, Cassandra y Redis. Una de sus características más importantes es la exploración de Big Data sin extracción, transformación y carga (ETL). Tiene la capacidad de generar informes y paneles HTML potentes de forma interactiva y directa desde grandes almacenes de datos [66].

3.9. Productos de proveedores para BDA.

Se presentan ejemplos de proveedores de software y hardware que ofrecen herramientas, plataformas y servicios para BDA en la Tabla 8.

Producto	Descripción
Cloudera	Es una distribución de software *open source* basado en *Apache Hadoop*. Para ayudar a las organizaciones a utilizar de forma fiable *Hadoop* en la producción, *Cloudera Enterprise* está específicamente diseñado para mejorar la capacidad de administración de las implementaciones de *Hadoop*, haciéndolo viable para usuarios corporativos, proporciona soporte técnico, actualizaciones, herramientas administrativas para clústeres *Hadoop*, servicios profesionales, capacitación y certificación [67].
Hortonworks	Proporciona plataformas de datos abiertas y aplicaciones modernas, impulsando la innovación en las comunidades de código abierto como *Apache Hadoop*, *NiFi* y *Spark*, las cuales potencian aplicaciones modernas que proporcionan una inteligencia accionable desde todos los datos: en movimiento y en espera [68].
EMC Greenplum	La base de datos de EMC *Greenplum* es conocida por su arquitectura de procesamiento masivo paralelo compartido (MPP), motor de flujo de datos paralelo de alto rendimiento. Recientemente, ha lanzado *Greenplum* HD (una distribución de *Hadoo*p lista para la empresa)[67].
IBM	Se destacan 3 productos: *Netezza* es una plataforma de bases de datos analíticas. IBM *InfoSphere BigInsights* es una oferta basada en *Hadoop* de IBM para requerimientos empresariales. IBM *InfoSphere Streams* es una plataforma para el procesamiento analítico en tiempo real, proporciona de forma única velocidad para BDA en datos estructurados y no estructurados [67].
Kognitio	Ofrece WX2, una plataforma de DB analítica implementada de 3 maneras: como una licencia sólo de software, como un dispositivo de almacén de datos completamente configurado que se ejecuta en hardware estándar del sector o bajo demanda, a través de los datos asequibles en *cloud data-warehousing-as-a-service* (DaaS) [67].
ParAccel	*ParAccel Analytic DataBase* (PADB) es una plataforma de base de datos analítica de procesamiento masivo paralelo (MPP) con columnas, con fuertes características de optimización y compilación de consultas, compresión e interconexión de redes. A través de los módulos de Integración *On Demand*, los usuarios pueden integrar PADB con otras plataformas, incluyendo Teradata y Hadoop [67].
SAS	SAS *Data Integration Studio* proporciona soporte para *Hadoop*, lo que permite a los especialistas de integración diseñar trabajos de integración mediante una interfaz gráfica que genera código *Pig*. SAS incluye transformaciones empaquetadas de *Hadoop* [67].
Tableau Software	*Tableau* es una plataforma analítica centralizada para el descubrimiento y exploración de datos, también se utiliza como una plataforma de BI de uso múltiple, aplicada a las necesidades empresariales o departamentales, combina tanto *Big Data* como datos de menor volumen) [22].
Teradata	Soporta EDW (*Enterprise Data Warehouse System*) proporciona escalabilidad y rendimiento rápido, tol cargas de trabajo mixtas simultáneas, como las de informes estándar, gestión de rendimiento, OLAP, análisis avanzado y en tiempo real o *streaming* de datos. Teradata adquirió *Aster Data* en la que ha recibido una patente sobre su integración SQL nativa con *MapReduc*e llamada SQL-*MapReduce* [67].

Tabla 8.Productos de proveedores para BDA.

3.10. Síntesis

En este capítulo se presentó una breve descripción de las tecnologías BDA actuales y sus proveedores, con el fin de poder tener un breve conocimiento de sus funcionalidades principales y poder elegir las mejores herramientas para el procesamiento de Big Data en IoT enfocadas a las Cadenas de Suministro.

A partir de esta investigación, se entiende que cada plataforma de *Big Data* tiene su enfoque individual; algunos están diseñados para el procesamiento por lotes, mientras que otros son buenos para el análisis en tiempo real.

No existe duda de que los análisis de *Big Data* aún se encuentran en la etapa inicial de desarrollo, ya que las técnicas de *Big Data* existentes y las herramientas están muy limitadas para resolver completamente los problemas reales en este campo [1], por lo tanto, se deberían realizar más inversiones científicas por parte de gobiernos y empresas en este paradigma científico para capturar valores enormes de *Big Data*. Desde el hardware hasta el software, inminentemente se requiere un almacenamiento más avanzado y técnicas de entrada/salida, arquitecturas informáticas más favorables, tecnologías más progresivas (plataformas *Big Data* con arquitectura sólida, infraestructura, enfoque y propiedades) y datos intensivos más eficientes.

4. Solución propuesta de BDA en IoT para Cadenas de Suministro

La solución propuesta a desarrollar utiliza Python, la plataforma *open source* de *Big Data* CDH (*Cloudera Distributed Hadoop*), Lenguaje R y Tableau; la aplicabilidad de la investigación se demuestra con la ayuda de un escenario que implica el análisis de *Big Data* generados desde IoT y de una "Plataforma de Compra-Venta y Control de Stocks de Productos" para la gestión de transacciones involucradas en el manejo de Suministro.

El alcance de este marco está limitado a la generación, extracción, ingestión en HDFS, visualización y análisis de datos para el control de rotación de inventario que nos permita generar conocimiento y realizar la buena toma de decisiones. Este trabajo está destinado a facilitar el desarrollo de Cadenas de Suministro Inteligentes, enfocadas específicamente al control de Stocks de productos.

La solución tiene cuatro principales componentes de tecnología: Sensores IoT con tecnología RFID/NFC Modulo RC522, una Base de Datos de gestión de la Cadena de Suministro de una empresa mayorista, *Big Data* no estructurada generada a través de los Sensores de IoT RFID/NFC y por último la utilización de herramientas *open source* BDA.

En lugar de utilizar sensores físicos de IoT RFID/NFC, se utiliza un programa generador de registros Web, que simule la información dada por el sensor, al realizar alguna transacción o visita a la plataforma Web en tiempo real. Para este propósito, se ha implementado código Python.

El generador de registros Web proporciona una gran cantidad de datos no estructurados, que se envían al puerto TCP (Protocolo de Control de Transmisión) 2181, donde un agente configurado de Apache Flume se encuentra en ejecución escuchando en este puerto. Este agente está configurado con el generador de registros Web obteniendo flujos de datos en tiempo real como fuente y HDFS como sumidero, por lo que almacena los datos en HDFS después de haber estado escuchando desde el puerto 2181. Para BDA; se usan los datos no estructurados obtenidos del generador y se correlacionan con los datos estructurados de la Base de Datos de gestión de la Cadena de Suministro.

La plataforma CDH, permite el almacenamiento, análisis y visualización de datos facilitando el manejo de flujos de trabajo *Big Data* de extremo a extremo. Tableau y R se integran permitiendo ejecutar algoritmos de *Machine Learning* que ayuden a realizar predicciones y posteriormente obtener conocimiento para la buena toma de desiciones.

En este capítulo, se presenta el desarrollo y elaboración de la solución propuesta, tomando en cuenta los nuevos desafíos de BDA, la descripción detallada de su ejecución y funcionamiento.

4.1. Descripción general de CDH

CDH es la distribución más completa y probada de Apache Hadoop y proyectos relacionados. Ofrece almacenamiento escalable, procesamiento por lotes unificado, procesamiento de flujo, SQL interactivo, búsqueda interactiva, controles de acceso basados en roles y computación distribuida [61].

4.1.1. Funcionamiento de Impala con CDH

La solución Impala se compone de los siguientes componentes detallados a continuación:

i) Clientes

Se utiliza el cliente Hue e Impala Shell que interactúan con Impala. Estas interfaces se utilizan generalmente para enviar consultas o completar tareas administrativas.

ii) Hive Metastore

Almacena información sobre los datos disponibles para Impala. El *metastore* detecta las bases de datos disponibles y su estructura.

iii) Impala

Se ejecuta en *DataNodes*, coordina y ejecuta consultas. Cada instancia de Impala puede recibir, planificar y coordinar consultas desde los clientes.

iv) HBase y HDFS

Utilizados para el almacenamiento de los datos [61]. La Figura 23, ilustra cómo se posiciona Impala en el entorno de Cloudera:

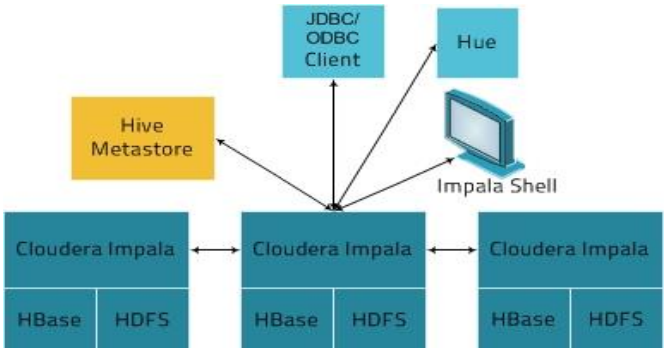


Figura 23. Posicionamiento de Impala en el entorno de Cloudera [61].

4.1.2. Manejo de consultas con Impala

Las aplicaciones de usuario envían consultas SQL a Impala a través de ODBC o JDBC, que proporcionan interfaces de consulta estandarizadas. La aplicación del usuario puede conectarse a cualquier Impalad. Servicios como HDFS y HBase son accedidos por instancias locales de Impala para proporcionar datos [61].

4.1.3. Características principales del Impala

A continuación, se detalla las características más importantes

- El Lenguaje Hive *Query Language* (HiveQL), incluye SELECT, y funciones agregadas.
- Almacenamiento en HDFS, HBase y *Amazon Simple Storage System* (S3), que incluye: Formatos de archivo HDFS: archivos de texto delimitados, Parquet, Avro, SequenceFile y RCFile.
- *Códecs* de compresión: Snappy, GZIP, Deflate, BZIP.
- Interfaces de acceso a datos comunes que incluyen: Controlador JDBC, Controlador ODBC, Hue Beeswax e Impala *Query UI.*
- Interfaz de línea de comandos *impala-shell.*
- Autenticación Kerberos [61].

4.1.4. Morphlines en la creación e integración de aplicaciones ETL para Hadoop

Cloudera Morphline es un nuevo marco de código abierto que reduce el tiempo y el esfuerzo necesario para integrar, crear y modificar aplicaciones de procesamiento de Hadoop que extraen, transforman y cargan datos en Apache Solr, Apache HBase, HDFS, EDW y paneles analíticos en línea[69].

Potencia una variedad de flujos de datos ETL de Apache Flume y MapReduce en Solr. Flume cubre el caso de tiempo real, mientras que MapReduce aborda procesamiento por lotes [69].

4.1.4.1. Modelo de procesamiento de Morphlines

Flume *Source* recibe eventos *syslog* y los envía a un Flume *Morphline Sink*, que convierte cada evento Flume en un registro; este extrae la línea de registro y lo canaliza a un comando *grok;* el cual usa una coincidencia de patrón de expresión regular para extraer algunas subcadenas de la línea; para luego conducir el registro estructurado resultante en el comando *loadSolr;* que finalmente carga el registro en Solr. En el proceso, los datos brutos o semiestructurados se transforman en datos estructurados acorde al modelado de la aplicación [69]. La Figura 24 muestra el esquema del modelo de procesamiento.

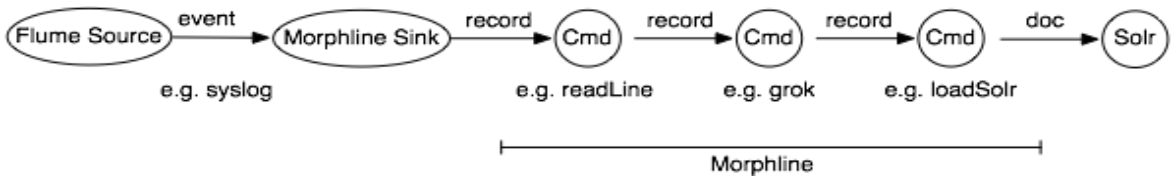


Figura 24. Modelo de procesamiento de Morphlines [69].

Esta integración permite la rápida creación de prototipos de aplicaciones ETL Hadoop, flujo complejo y procesamiento de eventos en tiempo real, análisis flexible de archivos de registro,

integración de múltiples esquemas de entrada heterogéneos y formatos de archivos, así como la reutilización de bloques lógicos ETL en aplicaciones Hadoop ETL [69].

4.2. Arquitectura de la solución propuesta

La arquitectura muestra los pasos involucrados en el proceso analítico. Como se discutió anteriormente, los datos de la Base de Datos de gestión de la Cadena de Suministro y datos provistos del generador de registros Web se ingieren en HDFS.

Desde HDFS, los datos se analizan utilizando Apache Hive, Impala y Spark; para la visualización de Datos se utiliza el cliente de Impala llamado Hue e Impala Shell que pueden interactuar con Impala, Tableau y R para las predicciones. Estas interfaces se utilizan generalmente para enviar consultas o completar tareas administrativas [61].

Hive *Metastore*: almacena información de los datos disponibles para Impala [61]. La información recopilada proporciona visibilidad detallada de los artículos enviados desde el fabricante a un minorista; permitiendo a los administradores de la cadena de suministro predecir y tomar decisiones factibles para la organización. A continuación, se muestra en la Figura 25 , la arquitectura propuesta para BDA.

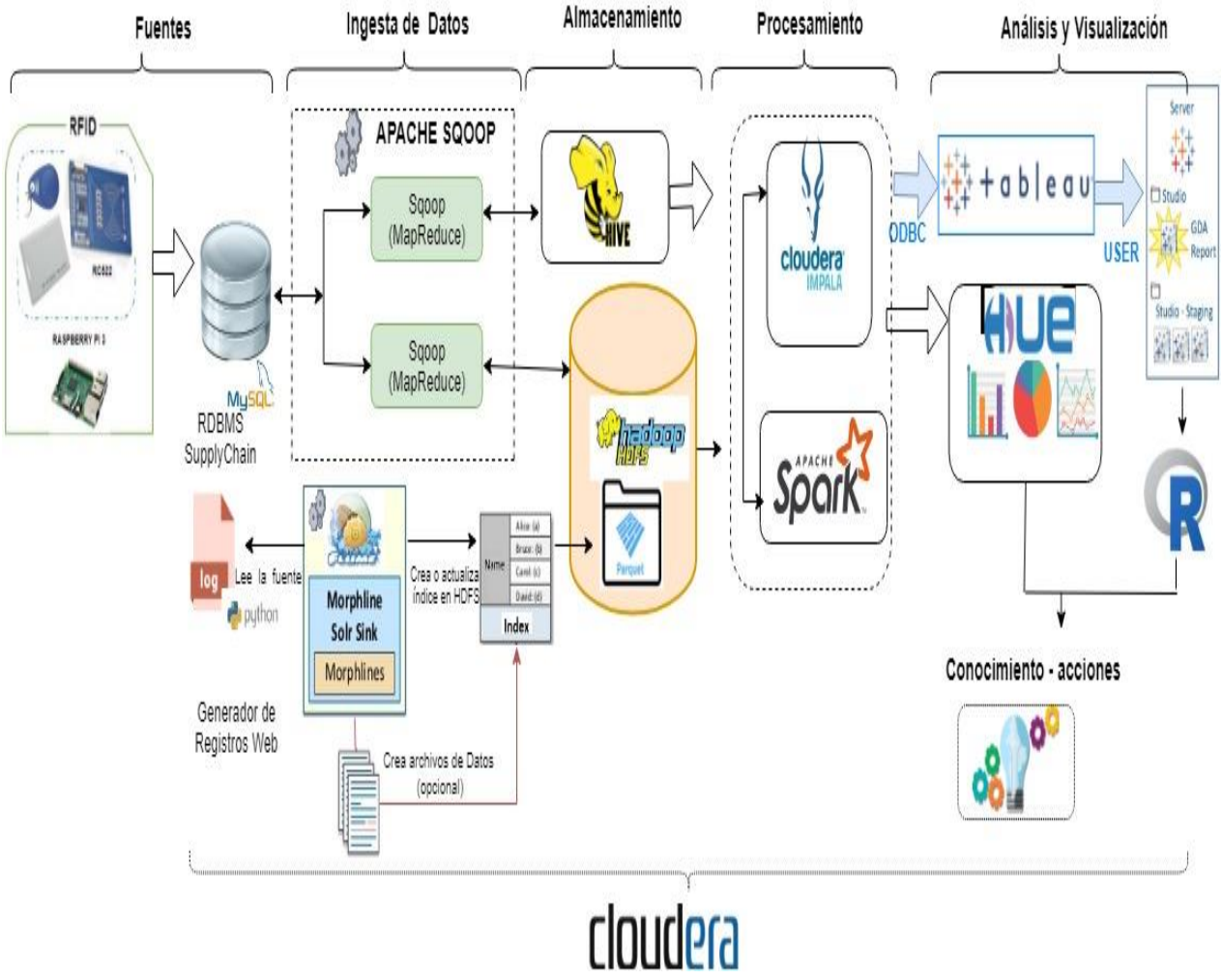


Figura 25. Arquitectura de BDA: ingerir, almacenar, analizar, visualizar y actuar.

4.3. Ingesta y consulta de datos relacionales en Hadoop

En el entorno RDBMS (*Relational Data Base Management System*) se visualiza los datos de varias transacciones en las que se indican cuáles son las categorías de los productos más comprados, los productos con mayor preferencia al comprar por los clientes y llevar un control de los datos generados por IoT en las Cadenas de Suministro.

Como punto de partida se restaura la Base de Datos de gestión de la Cadena de Suministro llamada **supplychaindb.sql;** que contiene información estructurada de miles de registros generados por una "Plataforma de Compra-Venta y Control de Stocks de Productos" mediante el uso de tecnología RFID, para después procesarlos dentro del entono CDH. En la Figura 26 se puede observar el modelo Entidad Relación.

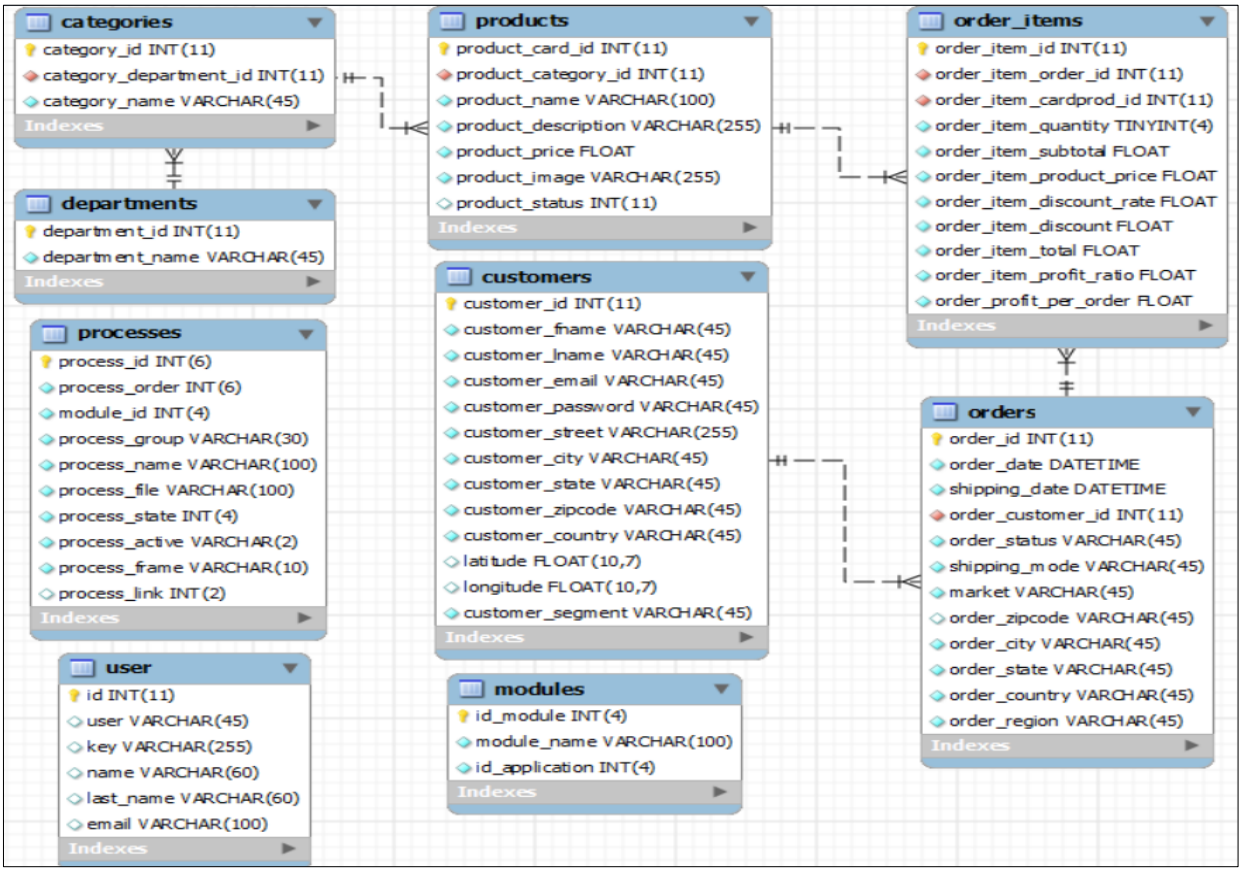


Figura 26. Modelo Entidad Relación de la Base de Datos supplychaindb.sql

Se ejecuta el siguiente comando para restaurar backup de la base de datos utilizando usuario: **root** y password **: cloudera** .

- `mysql -h localhost -u root -p supplychaindb < supplychaindb.sql`

Ingreso al servidor de Base de Datos MySQL con usuario: root y password : cloudera

- `mysql -h localhost -u root -p`

Ingreso a la base de datos supplychaindb a traves de los siguientes comandos :

- **show databases;**
- **use supplychaindb;**
- **show tables ;**

Se crea el usuario **supply_dba** otorgandole todos los privilegios . Para analizar los datos correspondientes a transacciones en la nueva plataforma, se debe transferir e incorporar los datos estructurados de un RDBMS a HDFS, mientras se preserva la estructura; permitiendo consultar los datos y interrumpir ninguna carga de trabajo regular con los mismos.

En CDH, se utiliza la herramienta del ecosistema de Hadoop: Apache Sqoop, la cual permite cargar automáticamente datos relacionales de MySQL a HDFS, conservando su estructura.

Se realiza la carga de estos datos relacionales directamente en un formulario listo para ser consultado por Apache Impala. Utilizando el formato de archivo Apache Avro, para la carga de trabajo en el clúster. En la terminal se inicia el trabajo en Apache Sqoop como se muestra en la Figura 27:

```
[root@quickstart ~]$ sqoop import-all-tables \
    -m 1 \
    --connect jdbc:mysql://quickstart:3306/supplychaindb \
    --username=supply_dba \
    --password=cloudera \
    --compression-codec=snappy \
    --as-parquetfile \
    --warehouse-dir=/user/hive/warehouse \
    --hive-import;
```

Figura 27. Inicio de trabajo con Apache Sqoop

El comando ejecutado lanza trabajos de MapReduce para extraer los datos de la Base de Datos **supplychaindb** y los escribe en HDFS; los mismos son distribuidos a través del clúster en formato de archivo Parquet. También el comando crea tablas para representar los archivos HDFS en Impala / Apache Hive con el esquema correspondiente. El procesamiento de trabajos MapReduce se muestra en la Figura 28.

Una vez completado el proceso, se confirma que los datos fueron importados correctamente a HDFS utilizando los siguientes comandos:

- **hadoop fs -ls /user/hive/warehouse/**
- **hadoop fs -ls /user/hive/warehouse/orders/**
- **hadoop fs -ls /user/hive/warehouse/products/**
- **hadoop fs -ls /user/hive/warehouse/categories/**

Figura 28. Procesamiento de trabajos MapReduce de creación de tablas y representación en archivos HDFS

Los comandos descritos muestran los directorios y dentro de estos, los archivos que componen las tablas de la base de datos **supplychaindb**, como se observa en la Figura 29.

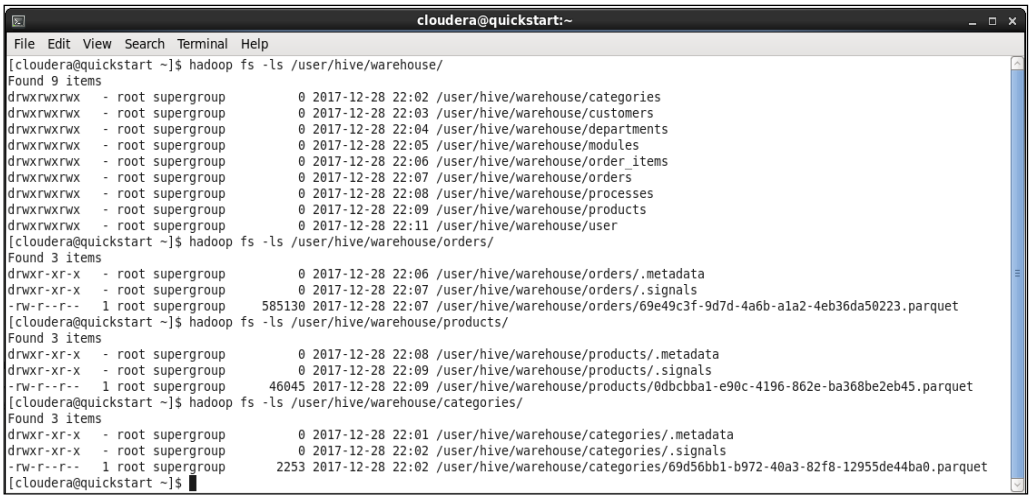


Figura 29. Consulta de directorios y archivos en HDFS

La cantidad de archivos en formato Parquet, mostrados es igual a la cantidad de mapeadores utilizados por Sqoop. En un nodo único, solo ve uno, pero los clústeres más grandes tienen una mayor cantidad de archivos.

La creación de tablas en Apache Hive e Impala se define mediante un esquema sobre los archivos existentes con sentencias *'CREATE EXTERNAL TABLE'*, similar a las Bases de Datos Relacionales. Para el procesamiento, visualización de datos y consulta de tablas se utiliza la aplicación Hue. El nombre de usuario administrador a utilizar para Hue es '**cloudera**' y su contraseña '**cloudera**', como se muestra en la Figura 30 en la siguiente dirección : http://quickstart.cloudera:8888/hue/.

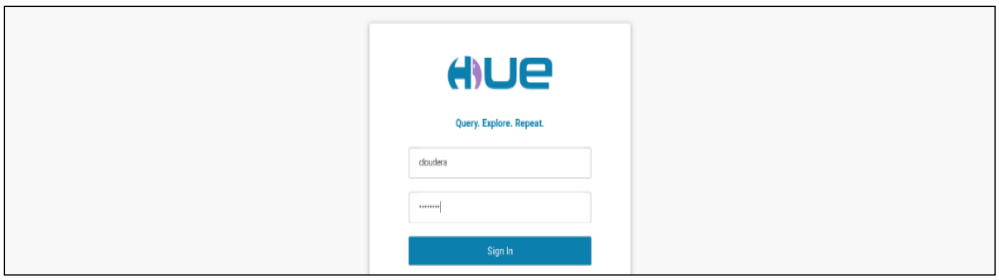


Figura 30. Interfaz Hue

Hue proporciona una interfaz basada en web para muchas de las herramientas en CDH y se encuentra en el puerto 8888 de su nodo administrador como se ilustra en la Figura 31.

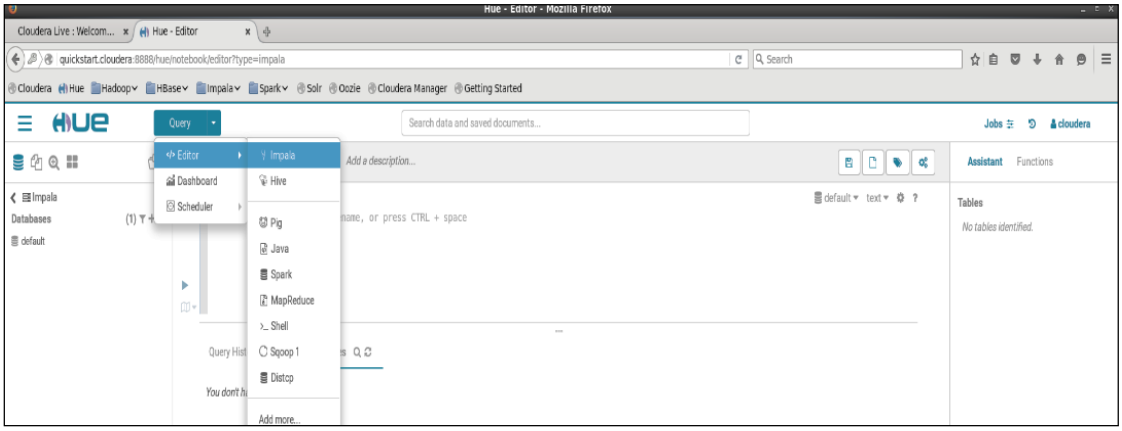


Figura 31. Inicio al Editor Impala

Para optimización de tiempo en ejecución de consultas y detectar cambios en los metadatos, se indica a Impala que sus metadatos están desactualizados ejecutando las sentencias:

- `invalidate metadata;`
- `show tables;`

Después hacer clic en el ícono "*Refresh Table List*" a la izquierda se observan las nuevas tablas en el menú lateral como muestra la Figura 32.

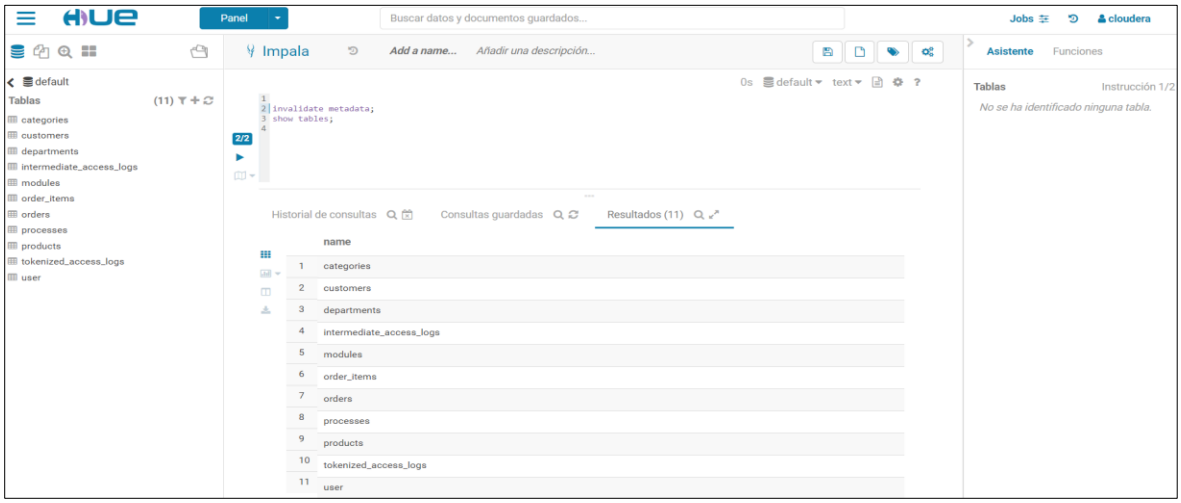


Figura 32. Visualización de tablas en HDFS

Ahora los datos de transacción se encuentran disponibles para consultas estructuradas en CDH. Se puede abordar cuestiones comerciales, por ejemplo, se consulta de los ingresos totales por categorías de productos como se muestra en la Figura 33.

```
--Categorías de productos más populares
select c.category_name, count(order_item_quantity) as quantity
from order_items oi inner join products p on oi.order_item_cardprod_id = p.product_card_id inner join categories c on c.category_id = p.product_category_id group by c.category_name order by quantity desc limit 15;
```

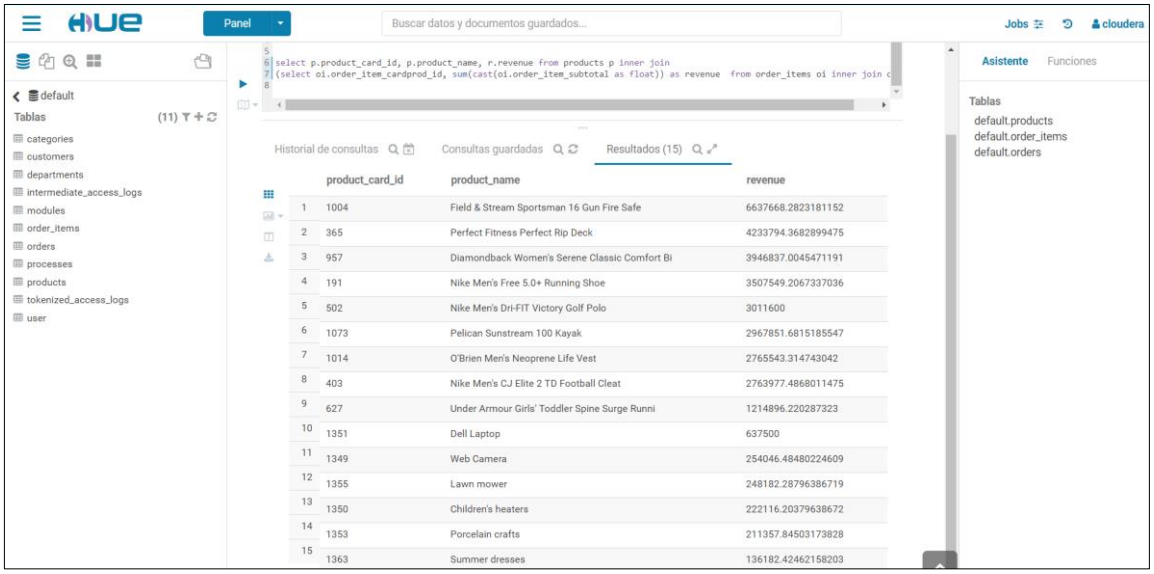
Historial de consultas | Consultas guardadas | Resultados (15)

	category_name	quantity
1	Cleats	24551
2	Men's Footwear	22246
3	Women's Apparel	21035
4	Indoor/Outdoor Games	19298
5	Fishing	17325
6	Water Sports	15540
7	Camping & Hiking	13729
8	Cardio Equipment	12487
9	Shop By Sport	10984
10	Electronics	3156
11	Accessories	1780
12	Golf Balls	1475
13	Girls' Apparel	1201
14	Golf Gloves	1070
15	Trade-In	974

Figura 33. Categorías de los productos más populares

Consulta los principales productos generadores de ingresos, como muestra la Figura 34 y la Figura 35.

```
-- Los 15 principales productos generadores de ingresos
select p.product_card_id, p.product_name, r.revenue from products p inner join (select oi.order_item_cardprod_id, sum(cast(oi.order_item_subtotal as float)) as revenue  from order_items oi inner join orders o on oi.order_item_order_id = o.order_id where o.order_status <> 'CANCELED' and o.order_status <> 'SUSPECTED_FRAUD' group by order_item_cardprod_id) r on p.product_card_id = r.order_item_cardprod_id order by r.revenue desc limit 15;
```



Historial de consultas | Consultas guardadas | Resultados (15)

	product_card_id	product_name	revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.2823181152
2	365	Perfect Fitness Perfect Rip Deck	4233794.3682899475
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837.0045471191
4	191	Nike Men's Free 5.0+ Running Shoe	3507549.2067337036
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967851.6815185547
7	1014	O'Brien Men's Neoprene Life Vest	2765543.314743042
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.4868011475
9	627	Under Armour Girls' Toddler Spine Surge Runni	1214896.220287323
10	1351	Dell Laptop	637500
11	1349	Web Camera	254046.48480224609
12	1355	Lawn mower	248182.28796386719
13	1350	Children's heaters	222116.20379638672
14	1353	Porcelain crafts	211357.84503173828
15	1363	Summer dresses	136182.42462158203

Figura 34. Principales productos generadores de ingresos

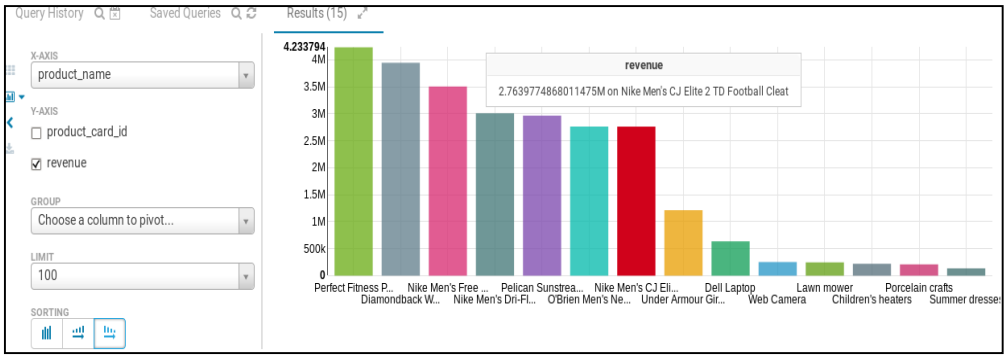


Figura 35. Diagrama de Barras de los principales productos generadores de ingresos

Sqoop importo los datos en Hive y se utilizó Impala para consultar los datos. Apache Hive e Impala pueden compartir los archivos de datos y los metadatos de la tabla. Hive trabaja compilando consultas SQL en trabajos de MapReduce, lo que lo hace muy flexible, mientras que Impala ejecuta consultas por sí mismo y está construido desde cero para ser lo más rápido posible, lo que lo hace mejor para el análisis interactivo.

4.4. Correlación de datos estructurados con datos no estructurados

Hadoop puede almacenar datos no estructurados y semiestructurados de gran volumen junto con datos estructurados sin remodelar una Base de Datos completa, también puede ingerir, almacenar y procesar eventos de registros web [53].

Una interrogante a nivel comercial que surge: **¿Son los productos más vistos o más buscados también los más vendidos?** Se puede descubrir qué visitantes de sitios realmente han visto más. Para este objetivo, se necesitan datos de secuencia de clics de la web. La forma más común de ingerir clics en la web es mediante la utilización de Apache Flume.

4.4.1. Datos *Clickstream* de carga masiva

Clickstream es la ruta que los visitantes eligen cuando navegan por un sitio web determinado. Es una colección de eventos recopilados en las páginas visitadas, los productos vistos, elementos buscados para un usuario determinado [70].

Para mayor rapidez en el análisis se realiza la carga de una muestra de 1877612 registros de datos de 5 meses, para lo cual se coloca la muestra en el directorio **/opt/examples/log_files/initial_access.log**. Estos datos son movidos a HDFS ejecutando los siguientes comandos desde el *Manager Node*:

- `sudo -u hdfs hadoop fs -mkdir /user/hive/warehouse/original_access_logs`
- `sudo -u hdfs hadoop fs -copyFromLocal /opt/examples/log_files/initial_access.log /user/hive/warehouse/original_access_logs`

Se verifica que los datos se encuentren en HDFS, ejecutando el siguiente comando: `hadoop fs -ls /user/hive/warehouse/original_access_logs`

A continuación, se crea una tabla en Hive y se consulta sus datos a través de Impala y Hue. Esta tabla se la crea en 2 pasos. Primero se toma ventaja de **SerDes** (serializadores / deserializadores) flexibles de Hive para analizar los registros en campos individuales utilizando una expresión regular. Segundo, se transfiere los datos de esta tabla intermedia a una que no requiera ningún SerDe especial. Una vez que los datos estén en esta tabla, se podrá consultar mucho más rápido e interactivamente usando Impala como se muestra en la Figura 36. En la aplicación Hive Query Editor en Hue se ejecuta las siguientes consultas:

```
CREATE EXTERNAL TABLE intermediate_access_logs (
    ip STRING,
    date STRING,
    method STRING,
    url STRING,
    http_version STRING,
    code1 STRING,
    code2 STRING,
    dash STRING,
    user_agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES ('input.regex' = '([^ ]*) - - \\[([^\\]]*)\\] "([^\ ]*) ([^\ ]*) ([^\
]*)" (\\d*) (\\d*) "([^"]*)" "([^"]*)"','output.format.string' = "%1$$s %2$$s %3$$s %4$$s
%5$$s %6$$s %7$$s %8$$s %9$$s")
LOCATION '/user/hive/warehouse/original_access_logs';
CREATE EXTERNAL TABLE tokenized_access_logs (
    ip STRING,
    date STRING,
    method STRING,
    url STRING,
    http_version STRING,
    code1 STRING,
    code2 STRING,
    dash STRING,
    user_agent STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/warehouse/tokenized_access_logs';
ADD JAR /usr/lib/hive/lib/hive-contrib.jar;
INSERT OVERWRITE TABLE tokenized_access_logs SELECT * FROM intermediate_access_logs;
```

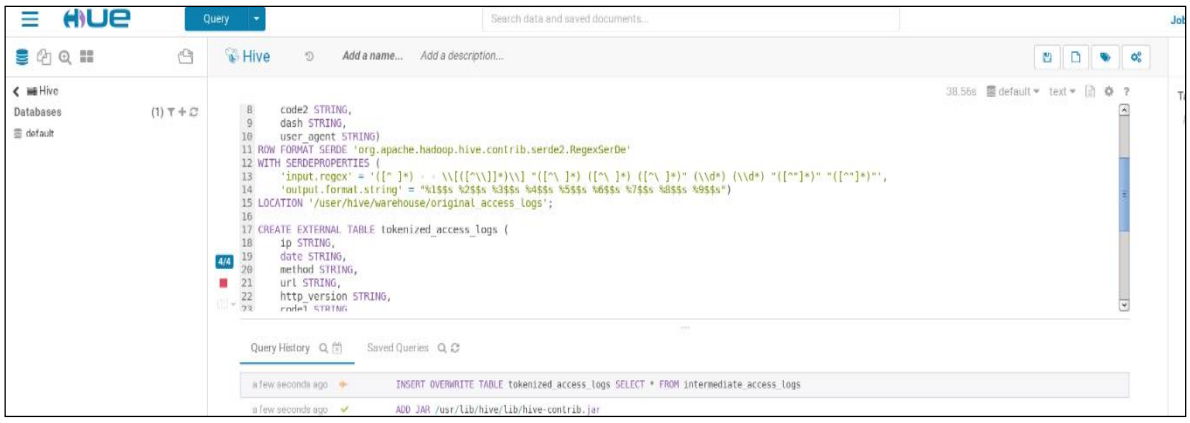


Figura 36. Transferencia de datos de una tabla a otra en paralelo con Hive

La consulta final utiliza un trabajo de MapReduce de transferencia de datos de una tabla a la otra en paralelo. Es necesario decirle a Impala que algunas tablas se han creado a través de una herramienta diferente. En la aplicación Impala Query Editor se ingresa el comando: **`invalidate metadata.`** Se visualiza 2 nuevas tablas externas en la base de datos predeterminada. Para observar los productos más vistos como se muestra en la Figura 37, se ejecuta la siguiente consulta:

```
-- Productos mas visitados
select count(*) as number_visits, url from tokenized_access_logs where url like
'%\/product\/%' group by url order by number_visits desc limit 10;
```

Figura 37. Productos más visitados.

4.4.2. El valor de *Big Data* en las búsquedas

Al introspectar los resultados, rápidamente se puede notar que esta lista contiene muchos de los productos que se encuentran en la lista de productos más vendidos mostrados en la Tabla 9, pero se puede observar que existe un producto que no apareció en el resultado anterior y es uno de los más vistos, pero nunca comprados, como indica la Tabla 10**.**

Ranking	***Product_card_id***	***Product_name***	***Revenue***
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.3
2	365	Perfect Fitness Perfect Rip Deck	4233794.4
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837
4	191	Nike Men's Free 5.0+ Running Shoe	3507549.2
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967851.7
7	1014	O'Brien Men's Neoprene Life Vest	2765543.3
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.5
9	627	Under Armour Girls' Toddler Spine Surge Runni	1214896.2
10	1351	Dell Laptop	637500

Tabla 9. Ranking de los productos más vendidos

Number_visits	**url**	***Product_ranking***
20258	/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck	**2**
18643	/department/apparel/category/featured%20shops/product/adidas%20Kids'%20RG%20III%20Mid%20Football%20Cleat	**?? *Missing***
18372	/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo	**5**
17963	/department/apparel/category/men's%20footwear/product/Nike%20Men's%20CJ%20Elite%202%20TD%20Football%20Cleat	**8**
11602	/department/fan%20shop/category/indoor/outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest	**7**
11577	/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak	**6**
11272	/department/fan%20shop/category/camping%20&%20hiking/product/Diamondback%20Women's%20Serene%20Classic%20Comfort%20Bi	**3**
10704	/department/fan%20shop/category/fishing/product/Field%20&%20Stream%20Sportsman%2016%20Gun%20Fire%20Safe	**1**
9958	/department/footwear/category/cardio%20equipment/product/Nike%20Men's%20Free%205.0+%20Running%20Shoe	**4**
9926	/department/footwear/category/fitness%20accessories/product/Under%20Armour%20Hustle%20Storm%20Medium%20Duffle%20Bag	**9**

Tabla 10. Ranking de los productos más visitados vs. los más vendidos

Esto sucedió debido a que, en la página de visualización, donde la mayoría de los visitantes se detenían, la ruta de ventas del producto tenía un error tipográfico en el precio del artículo, lo que ocasionaba una perdida en ventas. Existe riesgo de pérdida si una organización busca respuestas dentro de datos parciales. La correlación de dos conjuntos de datos mostró valor, hacerlo dentro de la misma plataforma facilita la gestión para una organización.

4.5. Análisis de fuerza de relaciones usando Apache Spark

Spark es una herramienta excelente para realizar un procesamiento de datos más avanzado: K-means, procesamiento de gráficos y ETL para trabajo pesado en tiempo real. Existe una semejanza entre Spark y MapReduce, para el ejemplo presentado, Spark usa conceptos muy similares de operaciones '*map*' y '*reduce*' (las operaciones '*join*' y '*groupBy*' son solo variaciones especiales de '*reduce*'). Sin embargo, la principal ventaja del uso de Spark es que el código es más conciso y los resultados intermedios se pueden almacenar en la memoria, lo que permite realizar secuencias complejas e iterativas mucho más rápido [51].

Se utiliza “Spark-on-YARN”, lo que significa que MapReduce y Spark comparten el mismo administrador de recursos, facilitando la administración del uso compartido de recursos entre muchos usuarios.

Se requiere posicionar los componentes juntos que generarán una fuerte cartera de clientes potenciales. A continuación, se realiza este trabajo en Spark y se da una idea de las relaciones entre productos. Para iniciar en la consola de Spark como se muestra en la Figura 38, ejecutamos el siguiente comando: **`spark-shell --master yarn-client`**

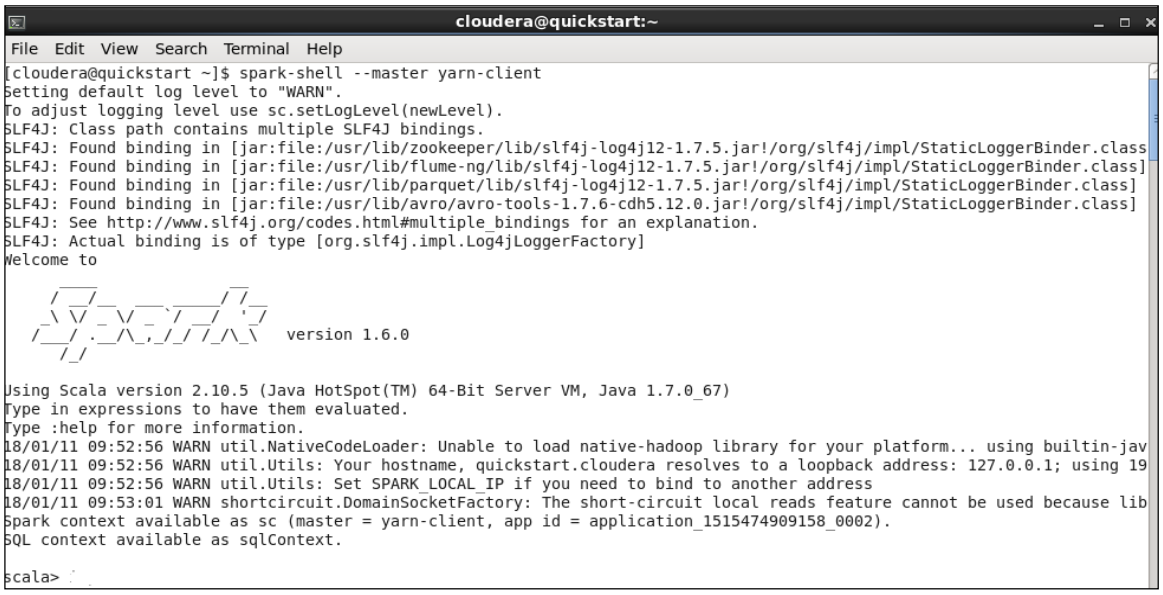


Figura 38. Ejecutando Spark en YARN

Una vez que aparece el indicador **scala>**, como se ilustra en la Figura 39, se ejecuta el código fuente Scala , que se presenta a continuación.

```
// Se importan las clases que necesarias
import org.apache.hadoop.mapreduce.Job
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat
import org.apache.avro.generic.GenericRecord
import parquet.hadoop.ParquetInputFormat
import parquet.avro.AvroReadSupport
import org.apache.spark.rdd.RDD

// Se crean RDD's para 2 de los archivos que se importó de MySQL con Sqoop
// Los RDD's son estructuras de datos de Spark para trabajar con conjuntos de
//datos distribuidos
def rddFromParquetHdfsFile(path: String): RDD[GenericRecord] = {
    val job = new Job()
    FileInputFormat.setInputPaths(job, path)
    ParquetInputFormat.setReadSupportClass(job,
        classOf[AvroReadSupport[GenericRecord]])
    return sc.newAPIHadoopRDD(job.getConfiguration,
        classOf[ParquetInputFormat[GenericRecord]],
        classOf[Void],
        classOf[GenericRecord]).map(x => x._2)
}
val warehouse = "hdfs://quickstart/user/hive/warehouse/"
val order_items = rddFromParquetHdfsFile(warehouse + "order_items");
val products = rddFromParquetHdfsFile(warehouse + "products");

// Extracción de los campos order_items y productos importantes
// y se obtiene una lista de cada producto,su nombre y cantidad agrupados por //orden
val orders = order_items.map { x => (
    x.get("order_item_cardprod_id"),
    (x.get("order_item_order_id"), x.get("order_item_quantity")))
}.join(
  products.map { x => (
    x.get("product_card_id"),
    (x.get("product_name")))
  }
).map(x => (
    scala.Int.unbox(x._2._1._1), // order_id
    (
        scala.Int.unbox(x._2._1._2), // quantity
        x._2._2.toString // product_name
    )
)).groupByKey()

// Por último, se calcula cuántas veces aparece cada combinación de productos
// juntos en un orden, luego los se ordena y se toma los 15 más comunes

val cooccurrences = orders.map(order =>
  (
    order._1,
    order._2.toList.combinations(2).map(order_pair =>
        (
            if (order_pair(0)._2 < order_pair(1)._2)
                (order_pair(0)._2, order_pair(1)._2)
            else
                (order_pair(1)._2, order_pair(0)._2),
            order_pair(0)._1 * order_pair(1)._1
        )
    )
  )
)
val combos = cooccurrences.flatMap(x => x._2).reduceByKey((a, b) => a + b)
val mostCommon = combos.map(x => (x._2, x._1)).sortByKey(false).take(15)
// Impresión de resultados, 1 por línea, y salida del shell Spark
println(mostCommon.deep.mkString("\n"))
exit.
```

Usando Spark y Scala, se pudo producir una lista de los artículos comprados con mayor frecuencia en muy poco tiempo.

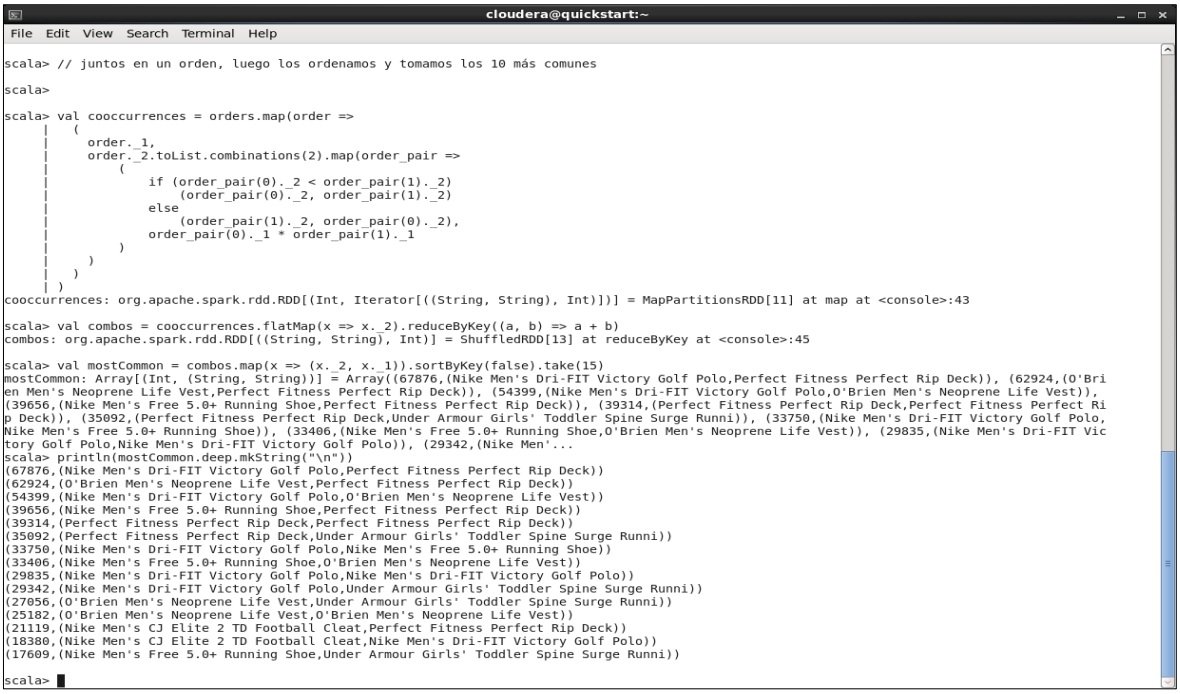


Figura 39. Análisis de coocurrencia con Spark y Scala

4.6. Ingesta de datos Clickstream del sitio web en tiempo real con Apache Flume

Se realiza la configuración de Apache Flume para ingesta escalable de datos de registro Web en tiempo real permitiendo: enrutar, filtrar, agregar y realizar "mini-operaciones" en los datos en su camino hacia la plataforma de procesamiento escalable CDH. El archivo flume.conf incluido en el Anexo A, explica a detalle los parámetros de configuración.

Flume lee los datos entrantes de una fuente específica (generador de registros web), esta información se procesa usando Morphlines. El índice de búsqueda se crea en HDFS y se actualiza a medida que llegan nuevos registros. Los datos procesados pueden escribirse opcionalmente como archivos en HDFS [69]. La Figura 40 , ilustra el esquema de Indexación con Flume.

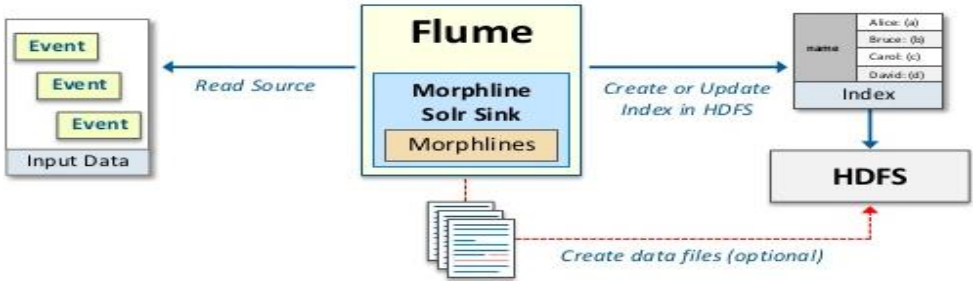


Figura 40. Indexación con Flume utilizando Morphlines. Adaptado de [69].

Los datos del registro web son registros estándar de un servidor web como se indica en la Figura 41, para luego publicar eventos en Apache Solr e indexarlos en tiempo real. Solr organiza estos datos de forma similar a la forma en que lo hace una base de datos SQL.

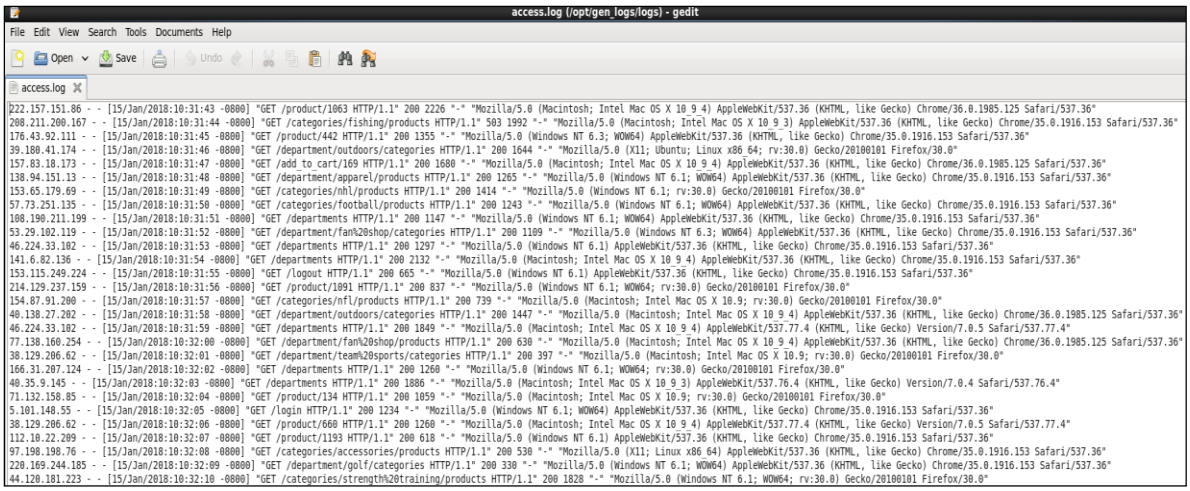


Figura 41. Datos del registro web

Se utilizó la interfaz de usuario de búsqueda Hue para la indexación en tiempo real por medio de Cloudera Search y Flume, de los datos de registro del servidor web de muestra.

4.6.1. Creación de índice de búsqueda con Apache Solr

Cuando se implementa un nuevo esquema de búsqueda, se lo realiza en cuatro pasos:

4.6.1.1. Creación de una configuración vacía

Se genera las configuraciones ejecutando el siguiente comando:

- **`solrctl --zk quickstart:2181/solr instancedir --generate solr_configs`**

La configuración y el archivo de esquema del clúster. Se pueden revisar explorando el directorio: **/opt/examples/flume/solr_configs.** Se adjunta la configuración del archivo morphline.conf , en el Anexo B. El resultado de este comando es una configuración básica que luego puede ser personalizada. Se personaliza el archivo principal **conf/schema.xml**, que se describe en el siguiente paso.

4.6.1.2. Edición de Esquema

El área más importante corresponde a la sección <fields> </ fields> del archivo **schema.xml**. Desde esta área se puede definir los campos que están presentes y que se pueden buscar en su índice como se presenta a continuación.

```
<field name="_version_" type="long" indexed="true" stored="true" multiValued="false" />
   <field   name="id"   type="string"   indexed="true"   stored="true"   required="true"
multiValued="false" />
   <field name="ip" type="text_general" indexed="true" stored="true"/>
   <field name="request_date" type="date" indexed="true" stored="true"/>
   <field name="request" type="text_general" indexed="true" stored="true"/>
   <field      name="department"      type="string"      indexed="true"      stored="true"
multiValued="false"/>
   <field name="category" type="string" indexed="true" stored="true" multiValued="false"/>
   <field name="product" type="string" indexed="true" stored="true" multiValued="false"/>
   <field name="action" type="string" indexed="true" stored="true" multiValued="false"/>
 </fields>
```

4.6.1.3. Carga de configuración de índice de búsqueda

Para realizar la carga se utilizan los siguientes comandos como se muestra en la Figura 42:

- `cd /opt/examples/flume`
- `solrctl --zk quickstart:2181/solr instancedir --create live_logs ./solr_configs`

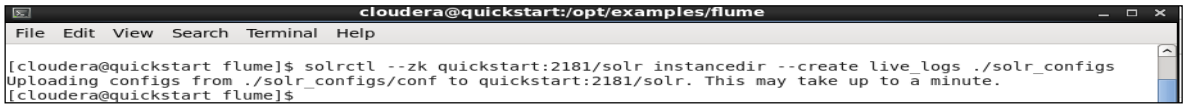


Figura 42. Carga de la configuración del índice de búsqueda

4.6.1.4. Creación de una "collection"

La creación se realiza ejecutando el siguiente comando:

- `solrctl --zk quickstart:2181/solr collection --create live_logs -s 1`

Para verificar la creación correcta de "*collection* " en Solr, se debe dirigir a Hue, después clic en Índices / colecciones. Se observa la colección que se terminó de crear, **live_logs**. La Figura 43, muestra los campos que fueron definidos, para el archivo schema.xml.

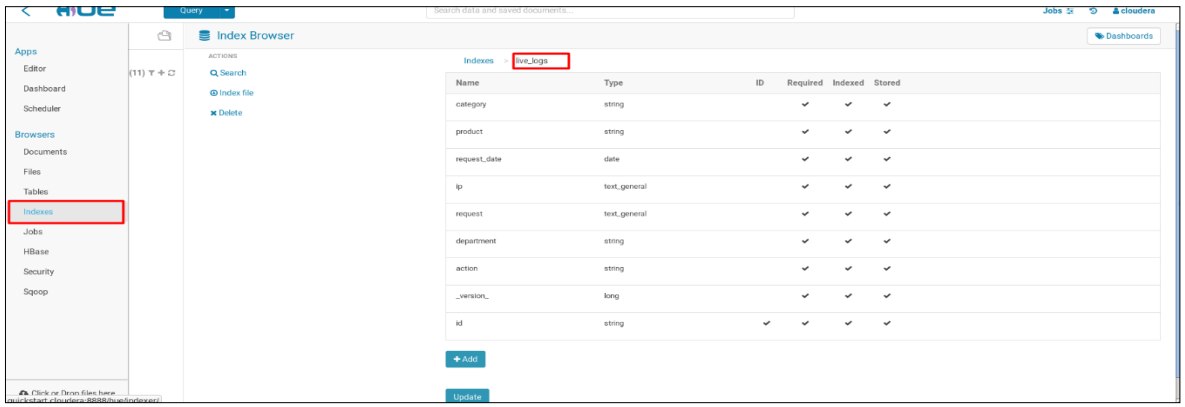


Figura 43. Creación de Collection en Solr

Se verifica la "*collection*" creada; el índice de búsqueda fue creado con éxito, ahora es posible comenzar a ingresar datos usando Flume y Morphlines.

Morphlines es una biblioteca de Java para hacer ETL (*on-the-fly*) es un excelente compañero de Flume [69]. Se define un Morphline que lee registros de Flume, los divide en campos que se requiere buscar y los carga en Solr. El ejemplo de Morphline presentado se define en **/opt/examples/flume/conf/morphline.conf**, es utilizado para indexar nuestros registros en tiempo real a medida que sean creados e ingeridos por Flume.

4.6.2. Ejecución del generador de registros Web

Para la ejecución se utiliza el programa Python Generador de Registros Web, su código fuente se encuentra adjunto en el Anexo C, con lo cual se obtiene datos de muestra como se observa en la Figura 44. Para iniciar su ejecución, se utilizan los siguientes comandos:

start_logs: Inicia de la ejecución del programa Python.
tail_logs: Finalizar con la ejecución y para regresar a su terminal se presiona <Ctrl+ C>.
stop_logs: Para detener la ejecución de los registros.

Figura 44. Ejecución del generador de registros web

4.6.3. Exploración de datos en tiempo real, utilizando Flume y Morphlines.

Una vez que se tiene un índice Solr vacío y eventos de registro en tiempo real que se ingresen al archivo **access.log** falso, se utiliza Flume y Morphlines para cargar el índice con los datos de registro en tiempo real.

Con pocos archivos de configuración simples, se utiliza Flume y un Morphline para realizar la carga de datos en el índice de Solr. Flume se utiliza para cargar muchos otros tipos de almacenes de datos; para este ejemplo se utiliza Solr. El inicio del agente de Flume se lleva a cabo ejecutando el siguiente comando:

```
[cloudera@quickstart ~]$ flume-ng agent \
    --conf /opt/examples/flume/conf \
    --conf-file /opt/examples/flume/conf/flume.conf \
    --name agent1 \
    -Dflume.root.logger=DEBUG,INFO,console ;
```

El agente de Flume empieza a ejecutarse en primer plano. Una vez que ha iniciado, empieza a procesar registros en tiempo real, como se visualiza en la Figura 45.

Figura 45. Procesamiento de registros con Agente de Flume

Al volver a la UI y hacer clic en "Buscar" en la página de la **"*Collection*"**, se puede buscar, profundizar y explorar los eventos que han sido indexados en tiempo real, como se visualiza en la Figura 46.

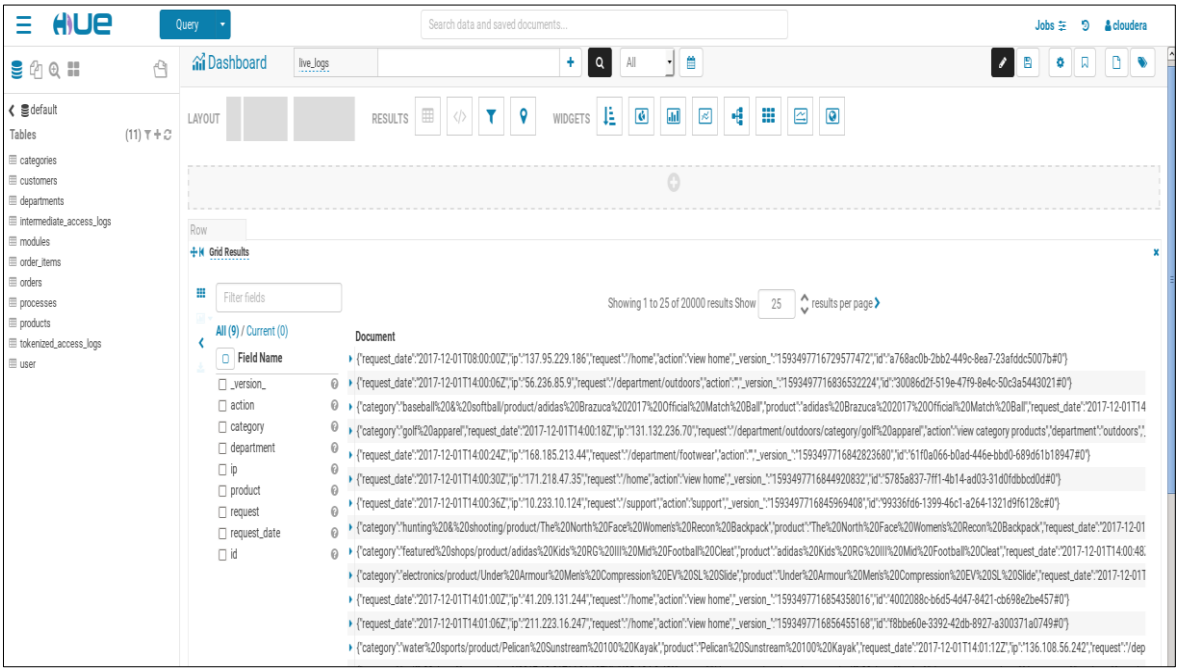


Figura 46. C*lickstreams*, indexados en una *Collection.*

En transcurso del proceso, se ha simulado la indexación de *clickstreams*, al mismo tiempo su ingesta a través de Flume a la plataforma, de modo que, si se escala un problema, se puede profundizar en los datos del último mes y explorar lo que sucede.

4.6.4. Construcción de un *Dashboard* en la plataforma HUE

Hue es un editor de consultas interactivas basado en la web en la pila Hadoop que permite visualizar y compartir datos [71]. La Tabla 11, muestra los componentes que son dependencias para las diferentes aplicaciones de Hue:

Componente	**Aplicaciones Dependientes**
HDFS	Core, File Browser
MapReduce	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
Hive	Hive Editor, Metastore Tables
Impala	Impala Editor, Metastore Tables
Pig	Pig Editor, Oozie
YARN	Job Browser, Job Designer, Oozie, Hive Editor, Pig, Sqoop
Oozie	Job Designer, Oozie Editor/Dashboard
Search	Solr Search
Spark	Spark
Sentry	Hadoop Security
Sqoop 2	Sqoop Transfer

Tabla 11. Dependencias de HUE [71].

En esta sección se describen los pasos a seguir para la elaboración de un *Dashboard* con la ingesta de *clikstreams* en tiempo real.

i) Colocar en modo de edición la herramienta HUE al seleccionar el icono de lápiz ubicado en la parte superior, donde puede elegir diferentes *widgets* y diseños, en este caso se eligió el grafico de barras;
ii) Se muestra la lista de campos que están presentes en el índice y se agrupa por el campo **request_date;**
iii) Se selecciona el *layout* de dos columnas, desde la esquina superior izquierda;
iv) Se arrastra un gráfico circular a la fila recién creada en la columna de la izquierda;
v) Se elige departamento como el campo de agrupación del grafico circular;
vi) La agregación de un filtro de faceta al lado izquierdo y se selecciona producto;
vii) Elegir el grafico de escala de tiempo, de acuerdo al campo **request_date**;
viii) Se arrastra un gráfico circular a la primera columna, agrupado por el campo **action;**
ix) Se realiza la selección del grafico de barras agrupado por el campo **request**;
x) Hacer clic en el ícono del lápiz para salir del modo de edición y finalmente se procede a guardar el *Dashboard* elaborado, como se presenta como en la Figura 47.

4.7. CDH, Lenguaje R y Tableau para BDA

Cloudera y Tableau son una poderosa solución de BDA. Los DWs multifacéticos y de rendimiento de Cloudera facilitan la tarea de almacenar y consultar *Big Data*. Tableau permite a los usuarios encontrar rápida y fácilmente información valiosa en vastos conjuntos de datos desde Hadoop, visualizarlos y crea cuadros de mando interactivos [72].

Tableau, incluye seguridad, gobierno y administración de nivel empresarial. Permite trabajar con datos de cualquier tamaño, almacenados en cualquier formato, en la nube, desordenados o perfectamente estructurados, accediendo de manera instantánea mediante extracción, conexión en vivo o una combinación de ambos [73].

R ofrece una poderosa manera de realizar análisis estadísticos y de minería en conjuntos de datos grandes. Las funciones y los modelos de R ahora pueden utilizarse en Tableau creando nuevos campos calculados que invoquen dinámicamente el motor de R y transmitan valores a este último [74]. La conexión y configuración de la fuente de datos con estas herramientas, se detallan en el Anexo D.

4.7.1. Descarga de *Drivers* requeridos

El conector de Tableau, requiere un controlador para poder conectarse con la fuente de datos alojada en CDH. Se solicita descargar 2 drivers:

- Cloudera ODBC *Driver* para Apache Hive;
- Cloudera ODBC *Driver* para Impala.

La descarga de los drivers requeridos e instrucciones de instalación, se encuentran en el enlace: https://www.tableau.com/support/drivers

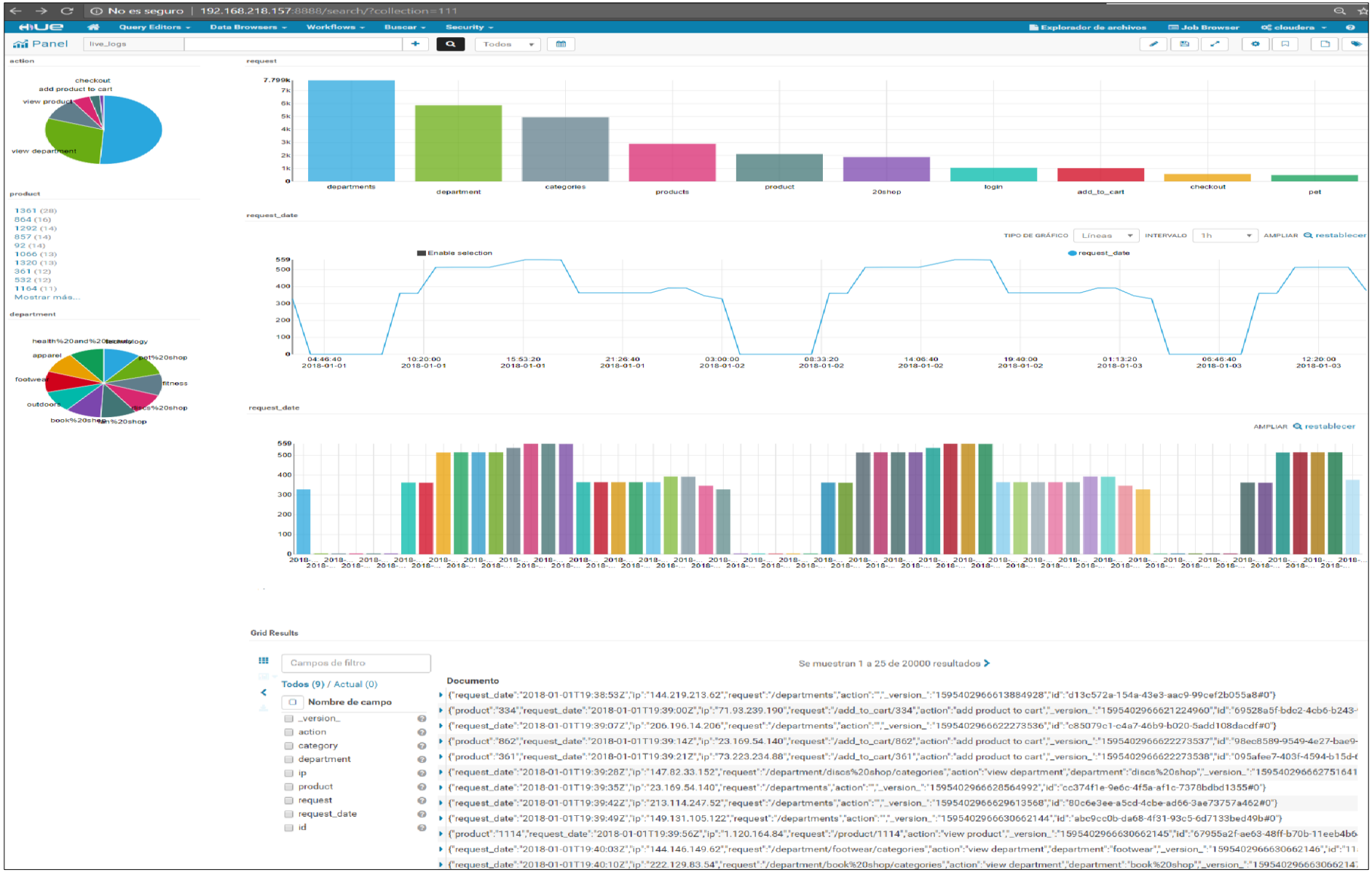


Figura 47. *Dashboard* de análisis de *clickstreams* en tiempo real.

4.7.2. Extracción de datos con Tableau

Para la extracción de datos, se arrastrastran las tablas de Hive correspondientes a *Supply Chain*, luego se configuran los *joins* de acuerdo a los campos de relación existentes. Al dar clic en "Actualizar ahora", presenta las 1000 primeras filas. La administración de Metadatos permite tareas de gestión rutinarias, como ocultar campos o actualizar nombres. Para la conexión a los datos activos, selecciónar "En tiempo real", como se muestra en la Figura 48.

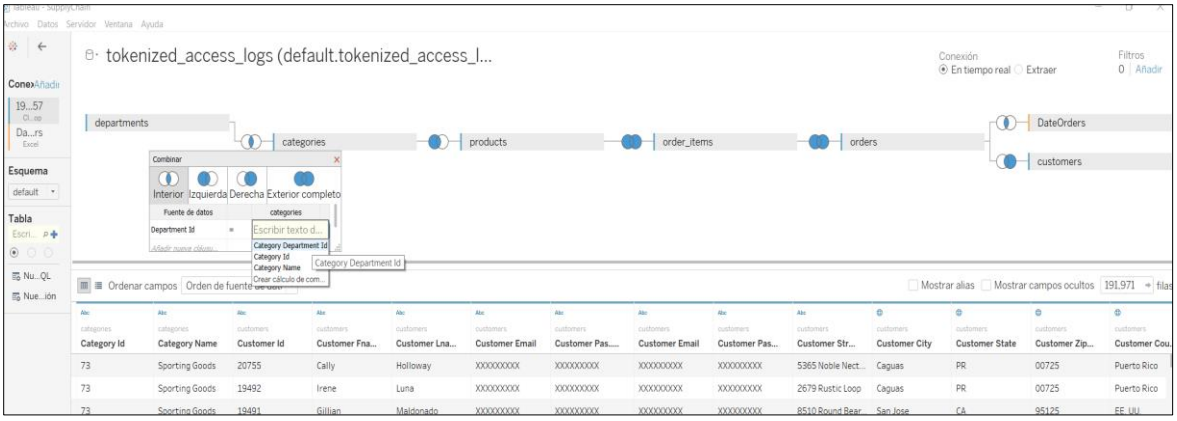


Figura 48. Extracción de Datos con Tableau.

4.8. Síntesis

En este capítulo se implementó la arquitectura propuesta, para el IoT y transacciones enfocadas al manejo de Suministro. Esto se logró también utilizando código Python para *clickstream* en HDFS.

Apache Sqoop permitió la importación de datos estructurados en HDFS, para más adelante convertirlo en formato de archivo Avro un formato de archivo muy utilizado en Hadoop, para después poder ser consultados en Impala, permitiendo así realizar informes donde la arquitectura de Hadoop frente a la de los sistemas tradicionales proporciona una escala y flexibilidad mucho mayores. Con la ayuda de Spark, y el Lenguaje Scala, se realizó el análisis de coocurrencia con alta rapidez, produciendo una lista de los artículos comprados con mayor frecuencia en muy poco tiempo.

El manejo de CDH en general permitió la exploración de datos en tiempo real, utilizando Flume, Solr y Morphlines. HUE ayudo a la visualización de estos datos en tiempo real mediante un *Dashboard.*

Big Data presenta oportunidades y desafíos para las empresas y Hadoop es una herramienta necesararia y eficiente para abordar esta temática, ya que permite realizar análisis en datos no estructurados provenientes de los registros web de gran volumen, y correlacionar con datos estructurados, dando valor al menejo de *Big Data* en las búsquedas.

5. Análisis de resultados

En este capítulo se da a conocer resultados obtenidos de la investigación realizada a través de BDA y técnicas de *Machine Learning* aplicadas.

5.1. Detección de Fraude con Machine Learning

La cadena de suministro tiene una exposición al riesgo muy alta, debe observarse desde una perspectiva preventiva, es decir, actuar antes que sucedan ese tipo de situaciones. El fraude en transacciones cuesta millones de dólares al año y evitarlas antes de que ocurran reduciría en gran medida los gastos de una organización [75].

Para el análisis y predicción de fraude, se utiliza Lenguaje R , seleccionando variables correspondientes a las transacciones incluidas en el *dataset*, como se muestra en la Tabla 12. El código fuente del programa R, para su pronóstico, se encuentra adjunto en el Anexo E.

Nombre de la variable	Tipo	Descripción
hour_month	<int>	Indica la hora del mes en que se capturaron estos datos, esta variable es calculada en base a la variable "*order_date"* .
type	<chr>	Indica el tipo de transacción realizada, es una variable categórica calculada en base a la variable "order_status"
Sales_per_customer	<dbl>	Muestra cantidad de dinero invertida en cada transacción
Customer_State	<chr>	Variable categórica que indica el lugar de origen de la transacción
Order_State	<chr>	Variable categórica que muestra el lugar de destino
isFraud	<int>	Variable categórica binaria calculada que indica 0: No hay Fraude , 1:Fraude.

Tabla 12.Seleccion de variables predictoras de fraude

Basado en el tipo de variable de resultado, se manejan cuatro modelos diferentes de *Machine Learnig*: Modelo Rpart, Modelo C5.0, Modelo *Random Forest* y SVM.Para determinar cuál se ajusta mejor al conjunto de pruebas en función de la curva ROC (Receiver Operating Characteristic) y la precisión. Cada modelo es probado contra 100.000 transacciones aleatorias categorizadas como "No Fraude", para estimar la tasa de falsos positivos. También se captura el tiempo de procesamiento estimando la intensidad computacional de cada modelo.

5.1.1. Procesamiento del conjunto de datos a modelar

Las columnas “type” e “isFraud” son categóricas y se cambian por factores. El atributo “isFraud”, que contiene los valores 0 y 1, es recodificada a: 0=” No”, 1 =” Yes”. A continuación, en la Figura 49 se presenta el resumen de transacciones fraudulentas y no fraudulentas.

```
isFraud            type          hour_month    Sales_per_customer name_dest_first
 No :176451   CASH    :20189   Min.   :  0.0   Min.   :   7.49    PR      :69373
 Yes:  4068   DEBIT   :69295   1st Qu.:182.0   1st Qu.: 104.38    CA      :29223
              PAYMENT :41725   Median :366.0   Median: 163.99     NY      :11327
              TRANSFER:49310   Mean   :365.2   Mean   : 183.11     TX      : 9103
                               3rd Qu.:548.0   3rd Qu.: 247.40     IL      : 7631
                               Max.   :743.0   Max.   :1939.99     FL      : 5456
                                                                   (Other):48406
```

Figura 49. Resumen de transacciones fraudulentas y no fraudulentas.

Se obtienen todas las transacciones que presentan fraude y se observa en la Figura 50, que el tipo de transacción “*PAYMENT”* no presenta fraude.

```
isFraud            type          hour_month    Sales_per_customer name_dest_first
 No :   0   CASH    : 574   Min.   :  0.0   Min.   :   8.66    PR      :1651
 Yes:4068   DEBIT   :   5   1st Qu.:184.0   1st Qu.: 104.38    CA      : 613
            PAYMENT :   0   Median :366.5   Median : 163.94    NY      : 217
            TRANSFER:3489   Mean   :360.2   Mean   : 185.06    IL      : 189
                            3rd Qu.:543.0   3rd Qu.: 248.96    TX      : 184
                            Max.   :739.0   Max.   :1939.99    FL      : 136
                                                               (Other):1078
```

Figura 50. Revisión de transacciones fraudulentas.

5.1.1.1. Reducción del conjunto de datos principal

Se eliminan variables insignificantes y se filtra por “CASH”, “DEBIT” y “TRANSFERS”. No existen transacciones por producto superiores a 1939.99, por lo que también es posible filtrar por este monto. La variable “hour_month”, indica la hora del mes en que se capturaron estos datos, por lo se consideran series temporales, como se muestra en la Figura 51.

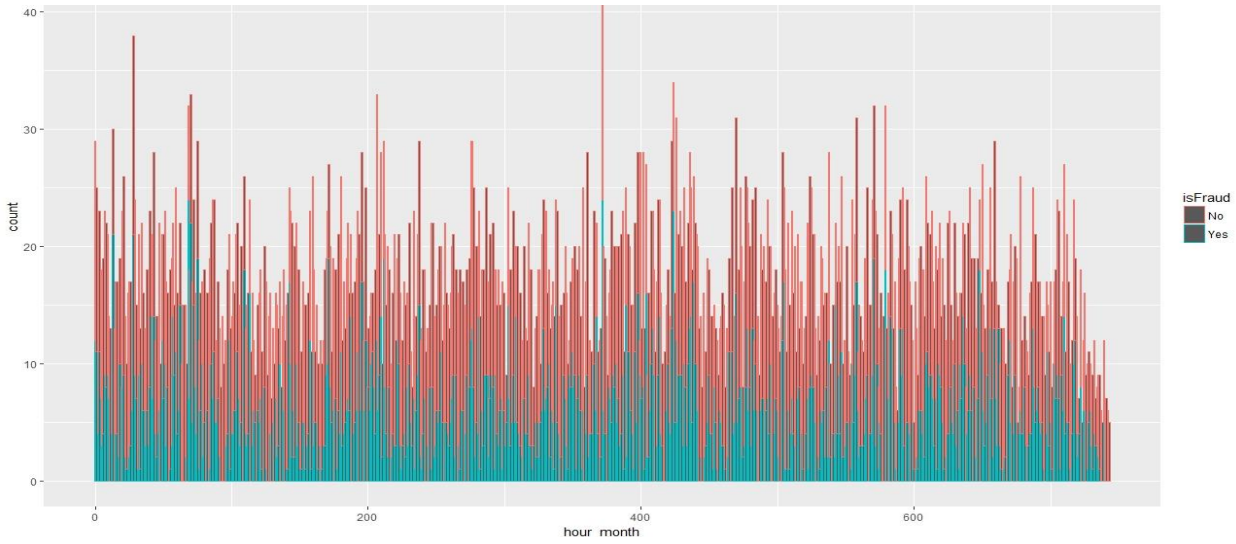


Figura 51. Series Temporales de transacciones Fraudulentas y No Fraudulentas

Existe correlación positiva entre variables “hour_month” y “Sales_per_costumer” de 0.0272. Para determinar la existencia de algún patrón de fraude, a continuación, se presenta el grafico de dispersión en la Figura 52, con un nivel de confianza del 90%.

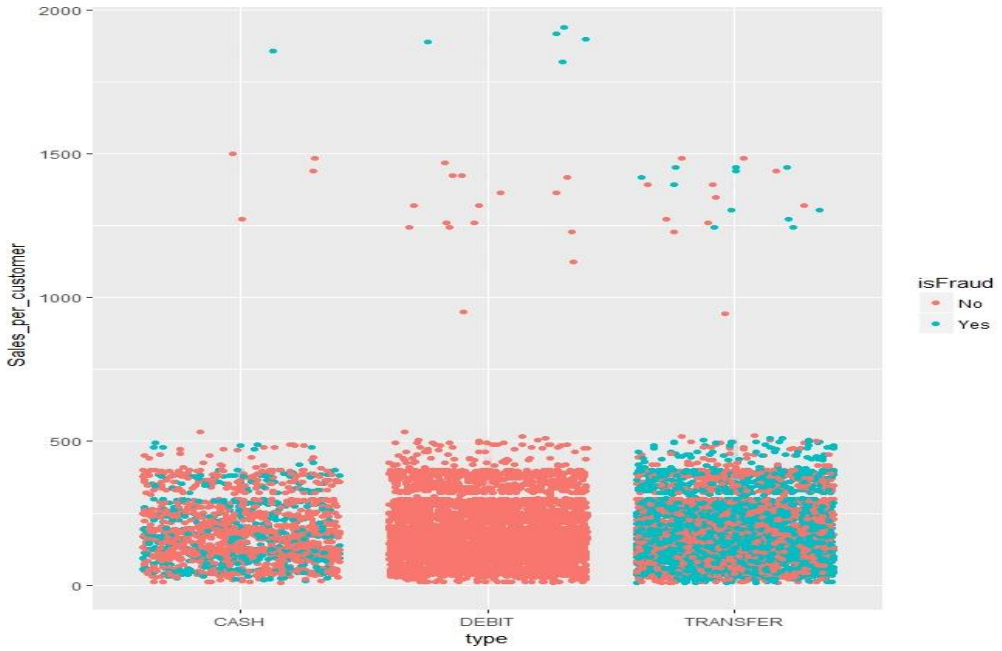


Figura 52. Búsqueda de patrones de fraude por monto de transacción.

Como se puede observar los puntos están por todos lados y aun no se determina un patrón. La Figura 53, muestra una síntesis del conjunto de transacciones preprocesadas.

```
isFraud           type         hour_month        Sales_per_customer
 No :134726   CASH    :20189   Min.   :-1.720470   Min.   :-1.4555
 Yes:  4068   DEBIT   :69295   1st Qu.:-0.867463   1st Qu.:-0.6527
              PAYMENT:    0    Median : 0.004395   Median :-0.1589
              TRANSFER:49310   Mean   : 0.000000   Mean   : 0.0000
                               3rd Qu.: 0.862114   3rd Qu.: 0.5330
                               Max.   : 1.781100   Max.   :14.5550
```

Figura 53. Síntesis del conjunto de transacciones preprocesadas.

No se observó la existencia de relaciones lineales entre predictores y se crea una muestra que contiene 9109 transacciones no fraudulentas y se incluye las 4068 transacciones no fraudulentas para el entrenamiento. Se crea un control para todos los modelos a evaluar utilizando tres iteraciones de validación cruzada de 10 veces para cada modelo para comparar su rendimiento y tiempo de respuesta computacional. Los modelos que se presentan en el apartado 5.1.2, sigue un patrón consistente, el cual es:

- Ajustar el modelo a los datos;
- Modelo comparado con los datos en los que fue entrenado;
- Modelo comparado con el conjunto de datos de prueba que se desconocía para la construcción del modelo;
- Comparación del modelo con una muestra de 100000 casos de **No Fraude** para determinar falsos positivos esperados.

5.1.2. Aplicación de modelos de *Machine Learning*

Este apartado se describe el uso de cada uno de los modelos mencionados en la Sección 5.1.

5.1.2.1. Modelo Rpart (Recursive Partitioning and Regression Trees)

Rpart construye modelos de clasificación o regresión. Los modelos resultantes se pueden representar como árboles binarios [76]. La Tabla 13, muestra resultados de la aplicación del modelo. El tiempo de entrenamiento fue de 6.699935 segundos.

Predicción en el conjunto de datos de entrenamiento

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  5306 1526
       Yes  875 2176
            Accuracy : 0.7571
              95% CI : (0.7485, 0.7655)
    No Information Rate : 0.6254
    P-Value [Acc > NIR] : < 2.2e-16
                  Kappa : 0.4625
 Mcnemar's Test P-Value : < 2.2e-16
            Sensitivity : 0.8584
            Specificity : 0.5878
         Pos Pred Value : 0.7766
         Neg Pred Value : 0.7132
             Prevalence : 0.6254
         Detection Rate : 0.5369
   Detection Prevalence : 0.6913
      Balanced Accuracy : 0.7231
       'Positive' Class : No
```

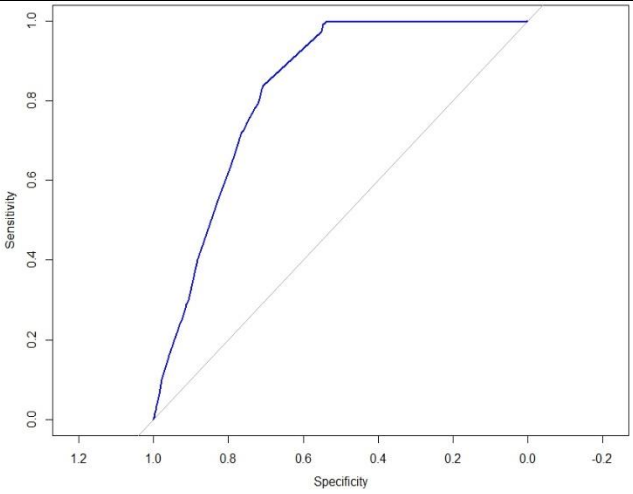
Predicción en el conjunto de datos de prueba

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1774  503
       Yes  327  690
               Accuracy : 0.748
                 95% CI : (0.7328, 0.7628)
    No Information Rate : 0.6378
    P-Value [Acc > NIR] : < 2.2e-16
                  Kappa : 0.4367
 Mcnemar's Test P-Value : 1.245e-09
            Sensitivity : 0.8444
            Specificity : 0.5784
         Pos Pred Value : 0.7791
         Neg Pred Value : 0.6785
             Prevalence : 0.6378
         Detection Rate : 0.5386
   Detection Prevalence : 0.6913
      Balanced Accuracy : 0.7114
'Positive' Class : No
```

Predicción en un gran conjunto de datos sin fraude

```
Time difference of 5.048236 secs
Confusion Matrix and Statistics
          Reference
Prediction    No   Yes
       No  76100 23900
       Yes     0     0
            Accuracy : 0.761
              95% CI : (0.7583, 0.7636)
    No Information Rate : 0.761
    P-Value [Acc > NIR] : 0.5017
                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16
            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.761
         Neg Pred Value :   NaN
             Prevalence : 0.761
         Detection Rate : 0.761
   Detection Prevalence : 1.000
      Balanced Accuracy : 0.500
'Positive' Class : No
```

Curva ROC contra los datos de prueba



Area under the curve: **0.8255**

Tabla 13. Resultados en R obtenidos del Modelo Rpart.

5.1.2.2. Modelo C5.0

El algoritmo C5 basado en Arboles de Decisión da el reconocimiento de ruido, falta de datos y puede anticipar qué atributos son relevantes y cuáles no en clasificación [77]. A continuación, se visualiza los resultados de la aplicación del modelo en la Tabla 14. El tiempo de entrenamiento del modelo fue de 2.532301 segundos.

<table>
<tr><th>Predicción en el conjunto de datos de entrenamiento</th><th>Predicción en el conjunto de datos de prueba</th></tr>
<tr><td>

```
  Confusion Matrix and Statistics

            Reference
  Prediction   No  Yes
         No  4468 2364
         Yes  441 2610

      Accuracy : 0.7162
        95% CI : (0.7072, 0.7251)
   No Information Rate : 0.5033
  P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.4338
Mcnemar's Test P-Value : < 2.2e-16
           Sensitivity : 0.5247
           Specificity : 0.9102
        Pos Pred Value : 0.8555
        Neg Pred Value : 0.6540
            Prevalence : 0.5033
        Detection Rate : 0.2641
  Detection Prevalence : 0.3087
     Balanced Accuracy : 0.7174
  'Positive' Class : Yes
```

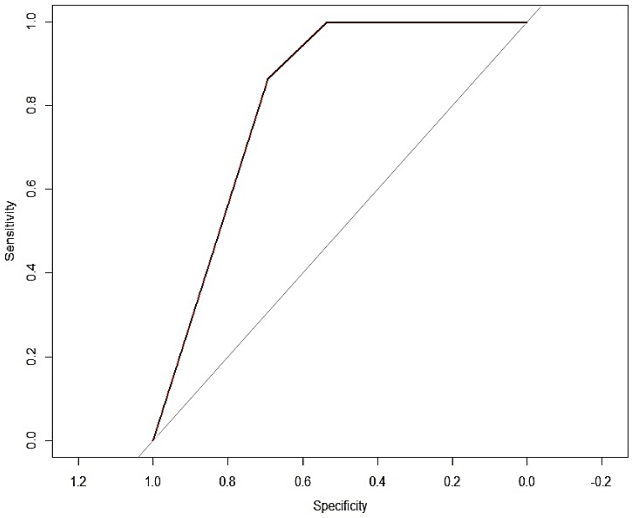
</td><td>

```
  Confusion Matrix and Statistics

            Reference
  Prediction   No  Yes
         No  1576  701
         Yes  138  879

      Accuracy : 0.7453
        95% CI : (0.7301, 0.7601)
 No Information Rate : 0.5203
P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.4323
 Mcnemar's Test P-Value : < 2.2e-16
           Sensitivity : 0.5563
           Specificity : 0.9195
        Pos Pred Value : 0.8643
        Neg Pred Value : 0.6921
            Prevalence : 0.4797
        Detection Rate : 0.2668
  Detection Prevalence : 0.3087
     Balanced Accuracy : 0.7379
      'Positive' Class : Yes
```

</td></tr>
<tr><th>Predicción en un gran conjunto de datos sin fraude</th><th>Curva ROC contra los datos de prueba</th></tr>
<tr><td>

```
Time difference of 4.110189 secs
  Confusion Matrix and Statistics

            Reference
  Prediction    No   Yes
         No  66055 33945
         Yes     0     0

      Accuracy : 0.6706
        95% CI : (0.6576, 0.6635)
No Information Rate : 0.6606
P-Value [Acc > NIR] : 0.5015

                   Kappa : 0
  Mcnemar's Test P-Value : <2e-16
            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.6605
             Prevalence : 0.3394
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000
     'Positive' Class : Yes
```

</td><td>



Area under the curve: **0.814**

</td></tr>
</table>

Tabla 14. Resultados en R obtenidos del Modelo C5.0.

5.1.2.3. Modelo Random Forests

Random Forests es un algoritmo de clasificación que combina árboles predictores tal que cada árbol depende de los valores de un vector aleatorio [78].

El error se nivela en alrededor de 100 árboles como se muestra en la Figura 54. Esto muestra que el Modelo *Random Forest* es un aprendiz efectivo en este conjunto de datos. Si estos datos contuvieran muchas más variables, se necesitarán más árboles, generalmente ocurren antes de 500 árboles. El modelo puede reducir de forma segura los árboles a 100 sin ningún impacto negativo significativo en el rendimiento.

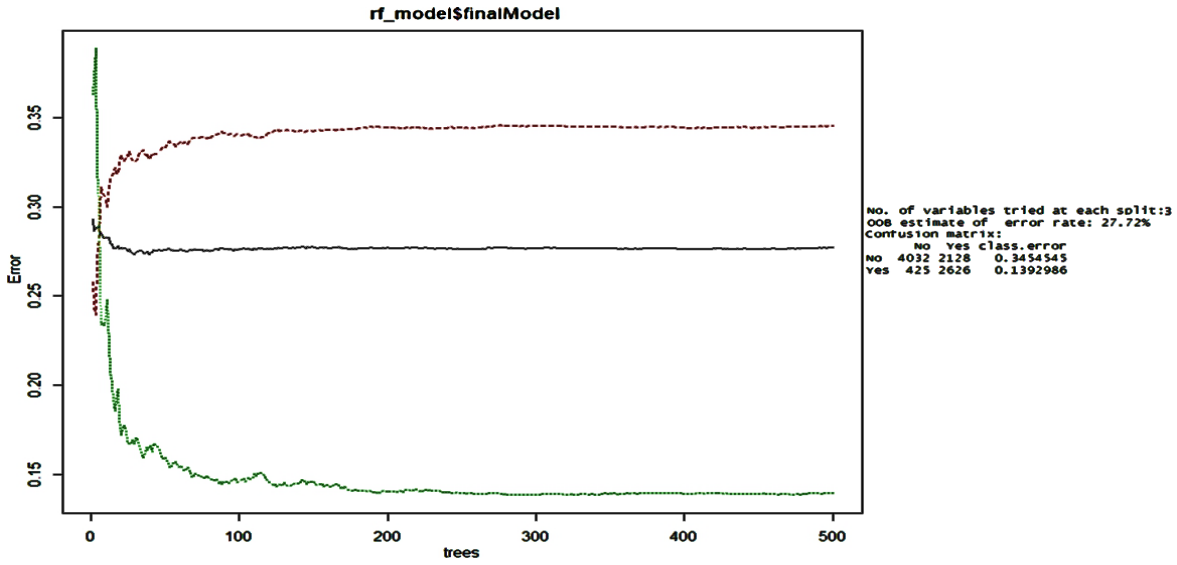


Figura 54. Bosques Aleatorios

La variable "typeDEBIT", contiene la información más significativa, "typeTRANSFER" y "hour_month" , también son influyentes , como se muestra en la Figura 55.

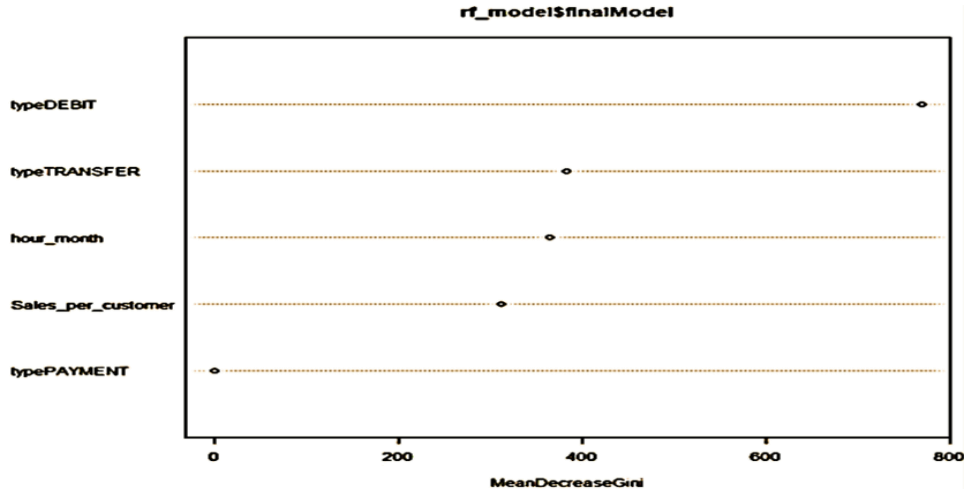


Figura 55. Importancia de las variables

La Tabla 15, presenta los resultados de la aplicación del modelo. El tiempo de entrenamiento es de 3.375788 minutos.

Predicción en el conjunto de datos de entrenamiento

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  6825    7
       Yes   22 3029

               Accuracy : 0.9971
                 95% CI : (0.9958, 0.998)
    No Information Rate : 0.6928
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9931
 Mcnemar's Test P-Value : 0.00933
            Sensitivity : 0.9977
            Specificity : 0.9968
         Pos Pred Value : 0.9928
         Neg Pred Value : 0.9990
             Prevalence : 0.3072
         Detection Rate : 0.3065
   Detection Prevalence : 0.3087
      Balanced Accuracy : 0.9972

       'Positive' Class : Yes
```

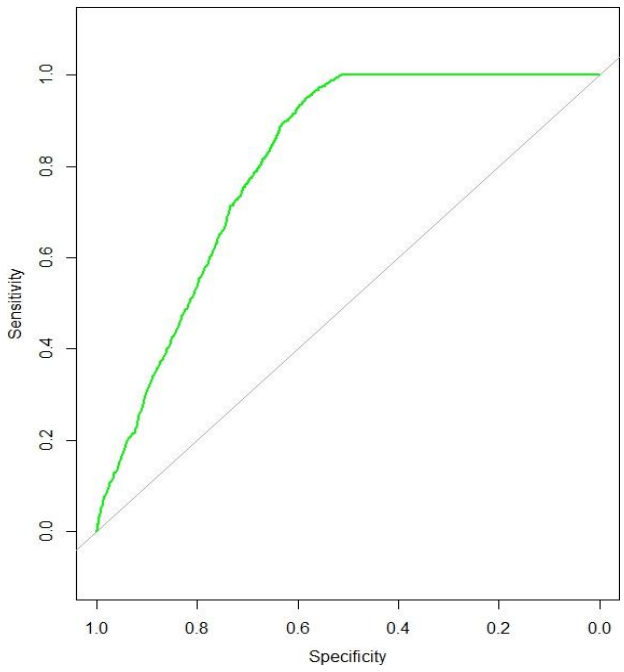
Predicción en el conjunto de datos de prueba

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1855  422
       Yes  457  560

               Accuracy : 0.7332
                 95% CI : (0.7177, 0.7482)
    No Information Rate : 0.7019
    P-Value [Acc > NIR] : 4.02e-05

                  Kappa : 0.3688
 Mcnemar's Test P-Value : 0.2515
            Sensitivity : 0.5703
            Specificity : 0.8023
         Pos Pred Value : 0.5506
         Neg Pred Value : 0.8147
             Prevalence : 0.2981
         Detection Rate : 0.1700
   Detection Prevalence : 0.3087
      Balanced Accuracy : 0.6863

       'Positive' Class : Yes
```

Predicción en un gran conjunto de datos sin fraude

```
Time difference of 4.616288  secs
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  81554 18446
       Yes     0     0

               Accuracy : 0.8155
                 95% CI :(0.8131, 0.8179)
    No Information Rate : 0.8155
    P-Value [Acc > NIR] : 0.502

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16
            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.8155
             Prevalence : 0.1845
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : Yes
```

Curva ROC contra los datos de prueba



Area under the curve: **0.8197**

Tabla 15.Resultados en R obtenidos del Modelo Random Forest.

5.1.2.4. Modelo SVM

Analiza los datos utilizados para la clasificación y el análisis de regresión , asignando nuevos ejemplos a una u otra categoría, convirtiéndolo en un clasificador lineal binario no probabilístico [26]. La Tabla 16, muestra resultados de la aplicación del modelo. El tiempo de entrenamiento del mismo fue de 19.4246 minutos.

Predicción en el conjunto de datos de entrenamiento

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  4779 2053
       Yes  710 2341

               Accuracy : 0.7204
                 95% CI : (0.7115, 0.7293)
    No Information Rate : 0.5554
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4161
 Mcnemar's Test P-Value : < 2.2e-16
            Sensitivity : 0.5328
            Specificity : 0.8707
         Pos Pred Value : 0.7673
         Neg Pred Value : 0.6995
             Prevalence : 0.4446
         Detection Rate : 0.2369
   Detection Prevalence : 0.3087
      Balanced Accuracy : 0.7017
       'Positive' Class : Yes
```

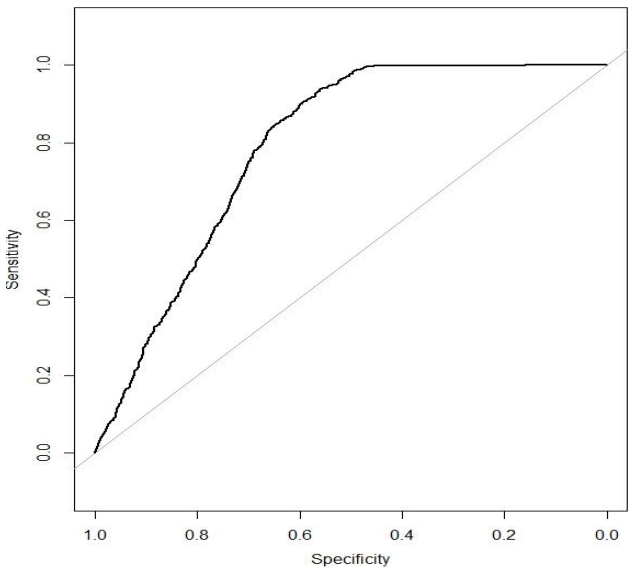
Predicción en el conjunto de datos de prueba

```
 Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1664  613
       Yes  238  779

               Accuracy : 0.7417
                 95% CI : (0.7263, 0.7565)
    No Information Rate : 0.5774
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4508
 Mcnemar's Test P-Value : < 2.2e-16
            Sensitivity : 0.5596
            Specificity : 0.8749
         Pos Pred Value : 0.7660
         Neg Pred Value : 0.7308
             Prevalence : 0.4226
         Detection Rate : 0.2365
   Detection Prevalence : 0.3087
      Balanced Accuracy : 0.7172
       'Positive' Class : Yes
```

Predicción en un gran conjunto de datos sin fraude

```
 Time difference of 38.09374  secs

Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  70217 29783
       Yes     0     0

               Accuracy : 0.7022
                 95% CI : (0.6993, 0.705)
    No Information Rate : 0.7022
    P-Value [Acc > NIR] : 0.5016

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16
            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.7022
             Prevalence : 0.2978
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000
       'Positive' Class : Yes
```

Curva ROC contra los datos de prueba



Area under the curve: **0.8129**

Tabla 16. Resultados en R obtenidos del Modelo SVM.

La Figura 56, presenta información importante resultante de la aplicación del modelo SVM.

```
Support Vector Machine object of class "ksvm"
SV type: C-svc  (classification) parameter : cost C = 1
Gaussian Radial Basis kernel function.
Hyperparameter : sigma =  0.550048198814586
Number of Support Vectors : 5754
Objective Function Value : -5615.993
Training error : 0.283416
Probability model included.
```

Figura 56. Información resultante de la aplicación del modelo SVM

5.1.3. Comparación de curvas ROC (Receiver Operating Characteristic)

Para la elección del modelo, se recurre a las curvas de rendimiento de diagnóstico ROC, ya que es una medida global e independiente del punto de corte [79].

La elección se realiza mediante la comparación del Área Bajo la Curva (AUC), correspondiente a cada uno de los modelos en uso. La Figura 57 , presenta el trazo de cada curva ROC.

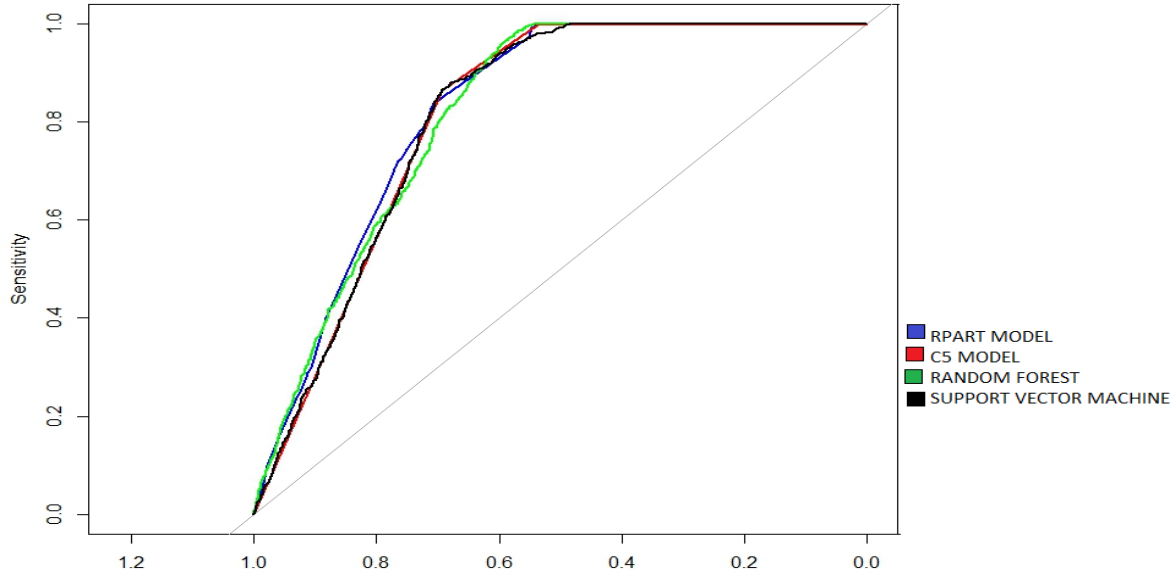


Figura 57. Comparación de curvas ROC.

Área debajo de las curvas ROC para cada modelo utilizaso es mostrado en la Tabla 17:

SVM	C5	RF	RPART
0.8128819	0.8139833	0.8196976	0.8255087

Tabla 17. Área debajo de las curvas ROC.

Los resultados indican que los modelos Random Forest y Rpart presentan un AUC bastante alto, para la elección del mejor modelo a utilizar, se realiza una evaluación con los resultados indicados en la Tabla 18 , correspondientes a las métricas utilizadas en grandes conjuntos de datos sin fraude.

Evaluación de modelos	Random Forest	Rpart
Accuracy	0.8155	0.761
False-positives	18446	23900
Run-time	4.616 seg	5.048 seg

Tabla 18. Métricas de elección de modelo.

Random Forest registra menor cantidad de falsos positivos y menor tiempo de ejecución en comparación a RPart, lo que significa que existe un 81,97% de probabilidad de que la predicción realizada a una transacción fraudulenta sea más correcta que el de una transacción normal escogida al azar.

5.2. Market Basket Analysis

Es un análisis matemático que ayuda a encontrar patrones en la información de las ordenes de venta de un conjunto de tiendas, durante un periodo determinado. La búsqueda de reglas de asociación con porcentajes de probabilidad indican qué artículos se suelen comprar al mismo tiempo, permitiendo conocer más sobre los hábitos de consumo de los clientes habituales [80]. El análisis se lo realiza utilizando Lenguaje R, seleccionando las siguientes variables, como se muestra en la Tabla 19. Su código fuente se adjunta en el Anexo F.

Nombre de la variable	Tipo	Descripción
order_Id	<int>	Indica el código de la orden .
product_Name	<chr>	Muestra el nombre del artículo, es una variable categórica.
order_item_quantity	<int>	Indica la cantidad de cada artículo ordenado .

Tabla 19. Selección de variables asociativas.

5.2.1. Pre procesamiento de datos y exploración

Se debe transformar los datos, del formato del marco de datos en transacciones de modo que se tengan todos los artículos comprados juntos en una fila. El conjunto de datos incluye 18059 registros y 3 campos. La variable "product_name" es categorizada como factor ; por lo tanto se observa que los clientes en su mayoría compran a lo más 5 artículos en cada orden de compra, como se indica en la Figura 58.

```
Observations: 180,519   Variables: 3
$ order_id            <int> 1, 2, 2, 2, 4, 4, 4, 4, 5, 5, 5, 5, 5, 7, 7, 7, 8, 8, 8, 8...
$ product_name        <fct> Diamondback Women's Serene Classic Comfort Bi, Nike Men's
Dri-FIT Victory Golf Polo, Pelican Sunstream 100 Kaya...
$ order_item_quantity <dbl> 1, 1, 5, 1, 5, 3, 2, 4, 5, 1, 1, 1, 2, 5, 1, 1, 5, 3, 1, 4...

>summary(retail)
    order_id                              product_name   order_item_quantity
 Min.   :    1   Perfect Fitness Perfect Rip Deck         :24515   Min.   :1.000
 1st Qu.:18057   Nike Men's CJ Elite 2 TD Football Cleat   :22246   1st Qu.:1.000
 Median :36140   Nike Men's Dri-FIT Victory Golf Polo      :21035   Median :1.000
 Mean   :36222   O'Brien Men's Neoprene Life Vest          :19298   Mean   :2.128
 3rd Qu.:54144   Field & Stream Sportsman 16 Gun Fire Safe:17325   3rd Qu.:3.000
 Max.   :77204   Pelican Sunstream 100 Kayak              :15500   Max.   :5.000
                 (Other)                                  :60600
```

Figura 58. Preprocesamiento de datos para *Market Basket Analysis*.

A continuación, se muestra en la Figura 59, la matriz de transacciones de artículos comprados en conjunto.

```
transactions as itemMatrix in sparse format with

 65752 rows (elements/itemsets/transactions) and
 118 columns (items) and a density of 0.02059137

most frequent items:

  Perfect Fitness Perfect Rip Deck 20359     Nike Men's CJ Elite 2 TD Football Cleat 18783
  Nike Men's Dri-FIT Victory Golf Polo 17869       O'Brien Men's Neoprene Life Vest 16623
  Field & Stream Sportsman 16 Gun Fire Safe 15164                           (Other) 70965

element (itemset/transaction) length distribution:
   sizes
    1     2     3     4     5
21174 14174 15102   11575  3727

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.00    2.00    2.43    3.00    5.00

includes extended item information               includes extended transaction information
      examples:                                                                examples:
                       labels                                          transactionID
1    adidas Brazuca 2014 Official Match Ball                      1               1
2      adidas Kids' F5 Messi FG Soccer Cleat                      2               2
3 adidas Men's F10 Messi TRX FG Soccer Cleat                      3               4
```

Figura 59. Síntesis de transacciones de artículos comprados en conjunto.

Se observa 65752 transacciones, y esta es la cantidad de filas, existen 118 elementos. Las transacciones aquí son las colecciones o subconjuntos de estos 118 artículos. Se puede calcular cuántos artículos se compraron usando la densidad como tal: (65752*118*0.02059137) = 159763. Se compraron 159763 artículos.

Se presenta el tamaño de las transacciones: 21174 transacciones fueron por solo 1 artículo, 14174 transacciones por 2 artículos, 1 transacción por 5 artículos. Esto indica que la mayoría de los clientes compran una pequeña cantidad de artículos en cada transacción. A continuación, se presenta la gráfica de frecuencia del artículo en la Figura 60.

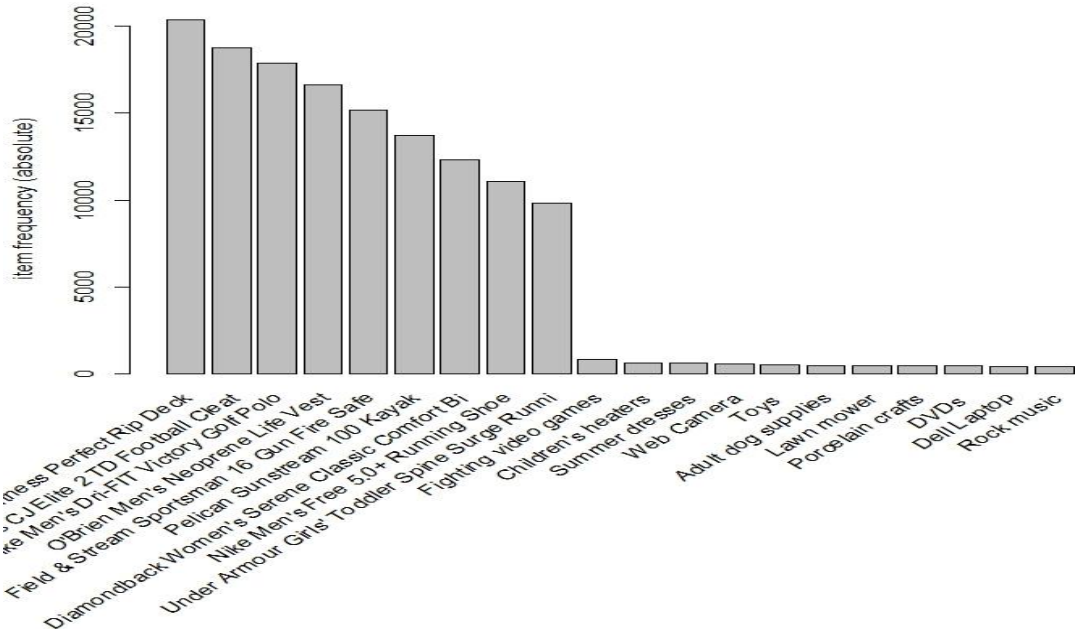


Figura 60. Frecuencia de compra de artículos

5.2.2. Creación de Reglas de Asociación

Se utiliza el modelo de asociaciones para encontrar relaciones entre distintos valores del dataset. Una asociación entre un antecedente y un consecuente. El uso del algoritmo Apriori en la biblioteca Arules de R, para extraer conjuntos de elementos frecuentes y Reglas de Asociación.

El algoritmo emplea búsqueda a nivel de conjuntos de elementos frecuentes. Con supp = 0.001 y conf = 0.35 para devolver todas las reglas que tienen un soporte de al menos 0.01% y una confianza de al menos 35% , como se indica en la Figura 61.

```
>inspect(rules)
 lhs                                                rhs                    support    confidence lift count
[1] {Glove It Women's Imperial Golf}             => {Perfect Fitness Rip Deck} 0.001779 0.386138 1.247084 117
[2] {Under Armour Women's Ignite PIP VI Slide}=>{Perfect Fitness Rip Deck} 0.001581 0.364912 1.178531 104
[3] {Bridgestone Straight Distance San Diego}=> {Perfect Fitness Rip Deck} 0.001688 0.359223 1.160158 111
[4] {LIJA Women's Eyelet Sleeveless Golf}    => {Perfect Fitness Rip Deck} 0.001688 0.359223 1.160158 111
[5] {Under Armour Men's Compression EV SL}   => {Perfect Fitness Rip Deck} 0.001520 0.358422 1.157573 100
[6] {Team Golf St. Louis Cardinals Putter}   => {Perfect Fitness Rip Deck} 0.001703 0.357827 1.155650 112
[7] {adidas Men's Germany Black Crest Away}  => {Perfect Fitness Rip Deck} 0.001566 0.356401 1.151044 103
[8] {ENO Atlas Hammock Straps}               => {Nike Men's CJ Football}   0.001733 0.355140 1.243208 114
[9] {LIJA Women's Eyelet Sleev. Golf Polo}   => {Nike Men's CJ Football}   0.001657 0.352750 1.234844 109
[10]{Clicgear Rovic Cooler Bag}              => {Perfect Fitness Rip Deck} 0.001520 0.352112 1.137193 100
[11]{ENO Atlas Hammock Straps}               => {Nike Men's Victory Golf } 0.001718 0.352024 1.295335 113
[12]{Titleist Pro V1x High Numbers Personali}=> {Nike Men's Victory Golf } 0.001627 0.350819 1.290900 107
[13{Glove It Women's Mod Oval 3-Zip Carry  } => {Perfect Fitness Rip Deck} 0.001566 0.350340 1.131468 103
[14]{Nike Men's Deutschland Weltmeister Win }=> {Nike Men's Victory Golf } 0.001581 0.350168 1.288504 104
```

Figura 61. Reglas de Asociación.

Según los resultados mostrados se observa que el 38.61% de las ordenes que contienen *"Glove It Women's Imperial Golf Glove"* también tienen *"Perfect Fitness Perfect Rip Deck"*, el 36.49% de órdenes que tienen *"Under Armour Women's Ignite PIP VI Slide"* también llevan *"Perfect Fitness Perfect Rip Deck".*"*Bridgestone e6 Straight Distance NFL*" es 1.16 veces más frecuente cuando *"Perfect Fitness Perfect Rip Deck"* aparece en una orden, además también se observa que*"LIJA Women's Eyelet Sleeveless Golf Polo"* es 1.16 veces más frecuente cuando *"Perfect Fitness Perfect Rip Deck"* aparece en una orden nueva.

5.2.3. Visualización de Reglas de Asociación

La mayoría de las reglas encontradas tienen muy bajo *Support*, ocurre debido a la gran cantidad de productos disponibles. La Figura 62 muestra los gráficos de dispersión para las 14 reglas presentadas de mayor confianza.

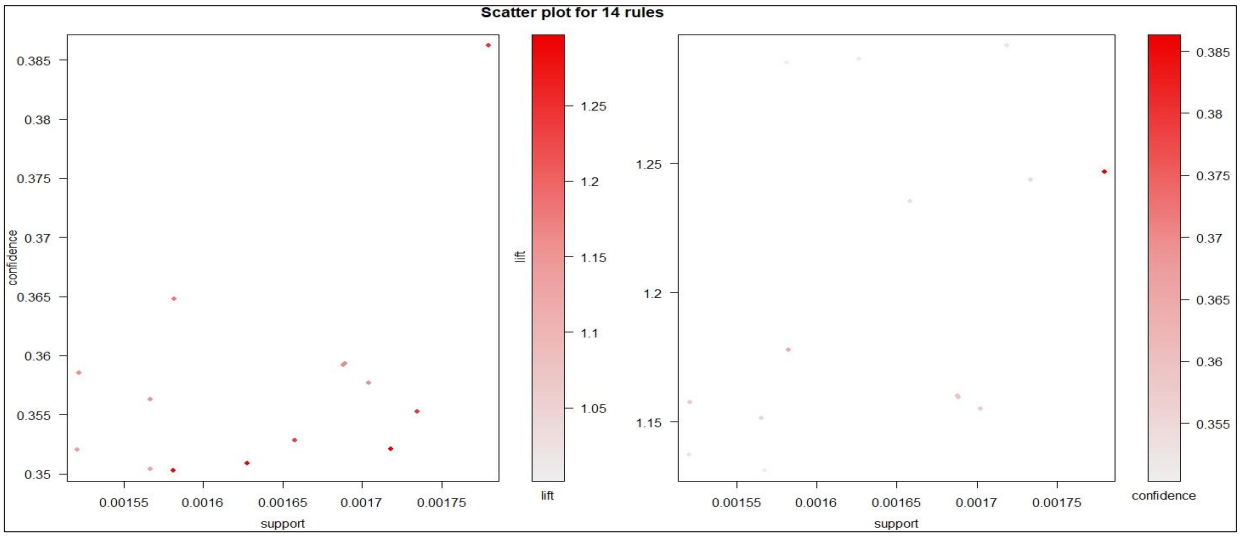


Figura 62. Gráficos de Dispersión de reglas principales.

La Figura 63, presenta las principales asociaciones de compra basadas en Grafos.

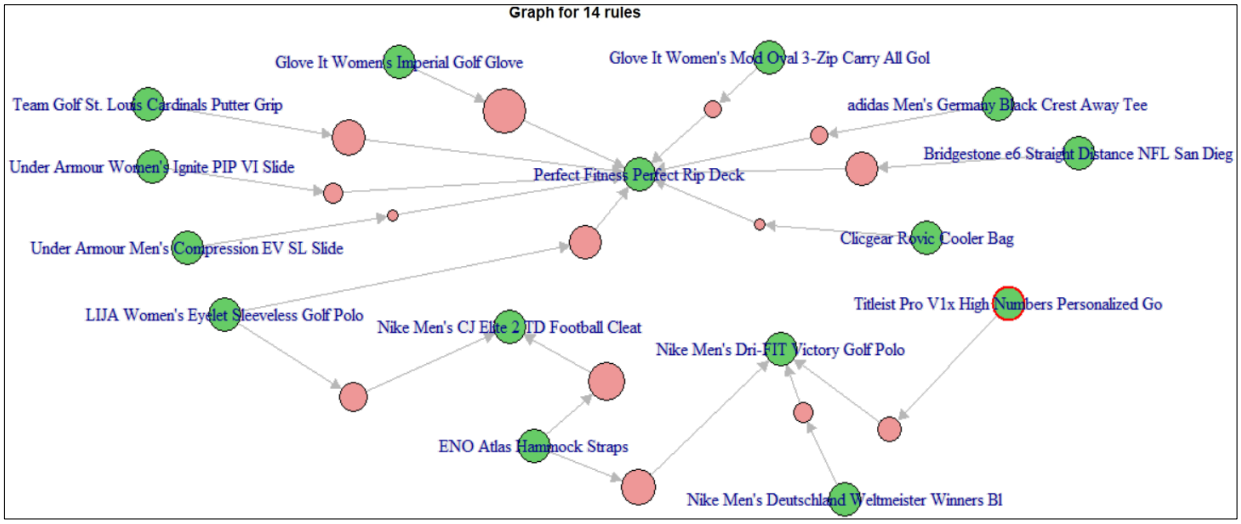


Figura 63. Visualización de Reglas basadas en Grafos

La Figura 64 , indica la descripción de artículos que conforman los Antecedentes y Consecuentes de las Reglas de Asociacion principales.

```
Itemsets in Antecedent (LHS)                                 Itemsets in Consequent (RHS)
 [1] "{Titleist Pro V1x High Numbers Personalized Go}"       [1] "{Perfect Fitness Perfect Rip Deck}"
 [2] "{Nike Men's Deutschland Weltmeister Winners Bl}"       [2] "{Nike Men's CJ Elite 2 TD Football}"
 [3] "{ENO Atlas Hammock Straps}"                            [3] "{Nike Men's Dri-FIT Victory Golf}"
 [4] "{Glove It Women's Imperial Golf Glove}"
 [5] "{LIJA Women's Eyelet Sleeveless Golf Polo}"
 [6] "{Under Armour Women's Ignite PIP VI Slide}"
 [7] "{Bridgestone e6 Straight Distance NFL San Dieg}"
 [8] "{Under Armour Men's Compression EV SL Slide}"
 [9] "{Team Golf St. Louis Cardinals Putter Grip}"
[10] "{adidas Men's Germany Black Crest Away Tee}"
[11] "{Clicgear Rovic Cooler Bag}"
[12] "{Glove It Women's Mod Oval 3-Zip Carry All Gol}"
```

Figura 64. Descripción de artículos en Antecedente y Consecuente.

A continuación, se detalla y se visualizan en la Figura 65 las reglas principales con mayor *Lift*.

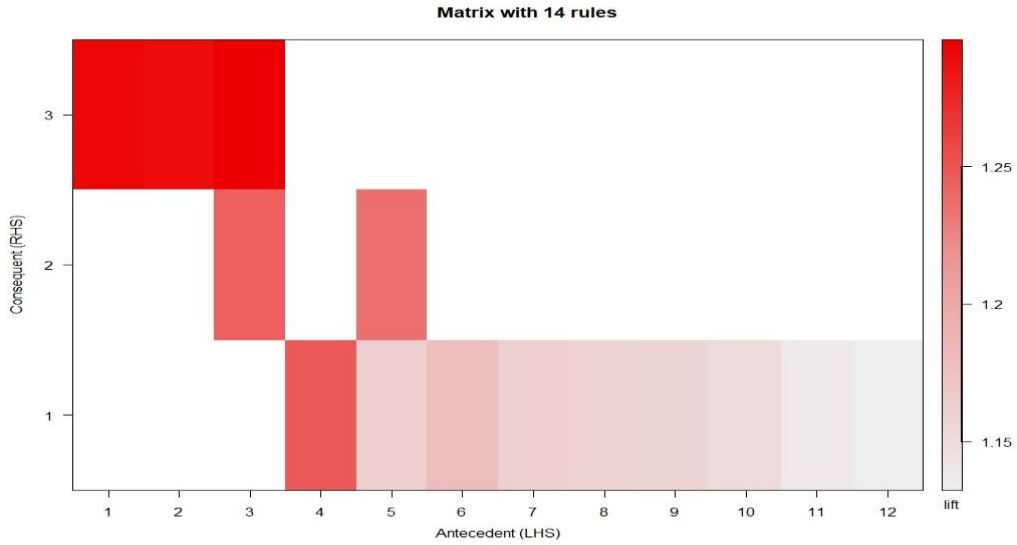


Figura 65. Visualización de Reglas en matriz

En la Figura 66, se puede notar que las compras más grandes tienen en general mayor cantidad de reglas, y con mayor *Support*, lo cual es esperable de la cantidad de productos aumentada.

Figura 66. Visualización de reglas agrupadas

5.3. Predicción de envíos tardíos con Regresión Logística

La regresión logística es un método para ajustar una curva de regresión, *y = f (x)*, cuando **"y",** es una variable categórica, y puede ser predecida por un conjunto de predictores *x, los* cuales pueden ser continuos, categóricos o una combinación de ambos [81].

Se utiliza el modelo de **"regresión logística binomial"** para clasificar y predecir si un envío es tardío (1) o no (0), ya que la variable para predecir es binaria. Para el análisis se utiliza el

software R, seleccionando las siguientes variables, como se muestra en la Tabla 20. El código fuente del programa, se encuentra adjunto en el Anexo G.

Nombre de la variable	Tipo	Descripción
Delivery Status	<fct>	Variable categórica que presenta el estado de un envió
Late_delivery_risk	<int>	Variable categórica binaria (Dependiente) que indica los estados 0: no tardío y 1 : tardío
Product Card Id	<int>	Indica el código de producto enviado
Order Item Quantity	<int>	Muestra el número de unidades por producto
Order Item Product Price	<dbl>	Presenta el precio por producto
Customer Segment	<fct>	Variable categórica que muestra el segmento del cliente
Order Status	<fct>	Variable categórica que indica el estado de la orden
Shipping Mode	<fct>	Variable categórica que indica el modo de envió de la orden

Tabla 20. Selección de variables predictoras de envíos tardíos.

5.3.1. Proceso de limpieza de datos

Se verifica si algunos datos pueden están perdidos o dañados, también cuántos valores únicos existen para cada variable. Para las variables categóricas, el uso de la función *read.csv()*, ayuda a codificarlas como factores. Un factor es cómo R trata las variables categóricas.

El diagrama de dispersión presentado en la Figura 67, permite para identificar envíos tardíos, agrupados por modo de envió de la orden con un nivel de confianza del 99%.

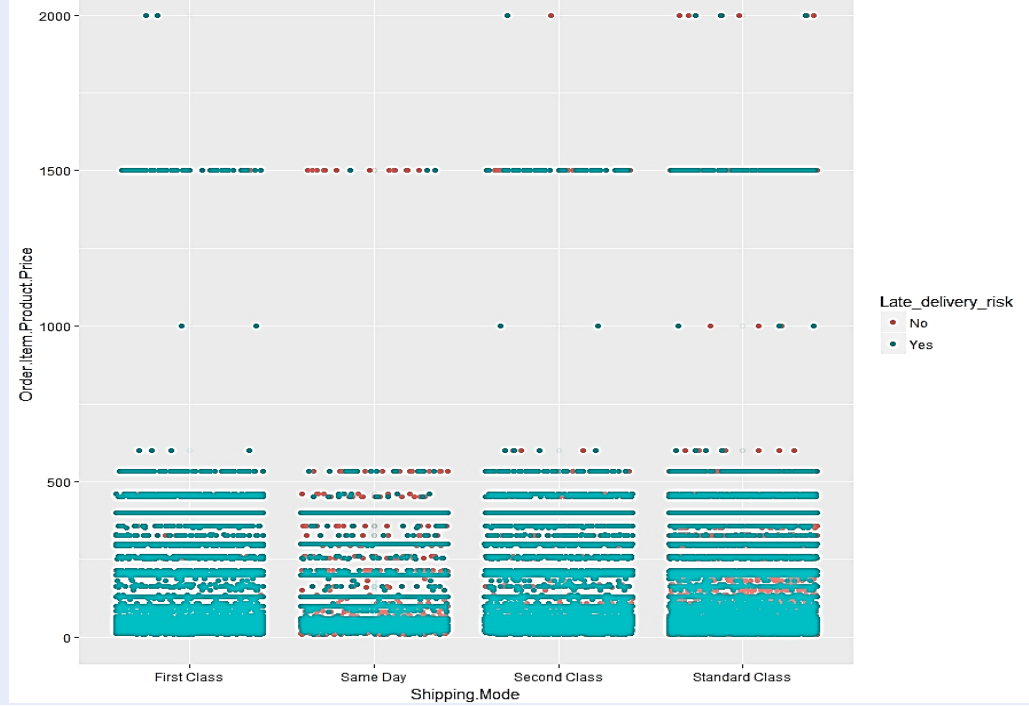


Figura 67. Diagrama de dispersión, envíos tardíos – modo de envió

R trata las variables categóricas, mediante la función *contrasts( )*. Esta función indica cómo las variables categóricas han sido manipuladas e interpretadas en el modelo,como se muestra en la Figura 68.

```
> contrasts(data$Shipping.Mode)
               Same Day Second Class Standard Class
First Class           0            0              0
Same Day              1            0              0
Second Class          0            1              0
Standard Class        0            0              1
```

Figura 68. Codificación de variables categóricas de envío.

5.3.2. Montaje y ajuste del modelo

El modelo es ajustado inicialmente con las variables presentadas en la Tabla 20, se indica su extracto en la Figura 69.

```
> summary(model)
Call:
glm(formula = Late_delivery_risk ~ ., family = binomial(link = "logit"),data = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.83014  -1.00760   0.00015   1.19441   1.52180
Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -1.621e+01  9.002e+01  -0.180   0.8571
Product.Card.Id              6.715e-05  2.756e-05   2.437   0.0148 *
Order.Item.Quantity         -5.455e-03  4.928e-03  -1.107   0.2683
Order.Item.Product.Price    -1.536e-04  7.040e-05  -2.182   0.0291 *
Customer.SegmentCorporate   -2.609e-02  1.417e-02  -1.840   0.0657 .
Customer.SegmentHome Office -9.011e-03  1.707e-02  -0.528   0.5975
Order.StatusCLOSED           3.442e+01  9.764e+01   0.353   0.7244
Order.StatusCOMPLETE         3.443e+01  9.764e+01   0.353   0.7243
Order.StatusON_HOLD          3.437e+01  9.764e+01   0.352   0.7249
Order.StatusPAYMENT_REVIEW   3.434e+01  9.764e+01   0.352   0.7251
Order.StatusPENDING          3.447e+01  9.764e+01   0.353   0.7241
Order.StatusPENDING_PAYMENT  3.446e+01  9.764e+01   0.353   0.7241
Order.StatusPROCESSING       3.443e+01  9.764e+01   0.353   0.7244
Order.StatusSUSPECTED_FRAUD  9.182e-03  1.241e+02   0.000   0.9999
Shipping.ModeSame Day       -1.831e+01  3.782e+01  -0.484   0.6284
Shipping.ModeSecond Class   -1.685e+01  3.782e+01  -0.445   0.6560
Shipping.ModeStandard Class -1.864e+01  3.782e+01  -0.493   0.6221
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 198848  on 144419  degrees of freedom
Residual deviance: 148414  on 144403  degrees of freedom
AIC: 148448
Number of Fisher Scoring iterations: 17
```

Figura 69. Extracto del modelo de Regresión Logística para envíos tardíos.

Al analizar con un nivel de confianza del 95%, existen variables que no son estadísticamente significativas como: “Order Status”, “Customer Segment”, “Order Item Quantity”, se procede a eliminar y reajustar nuevamente el modelo.

Los intervalos de confianza al 95%, son presentados para cada coeficiente estimado. A continuación en la Figura 70, se visualiza el montaje y ajuste del modelo de Regresión Logísitica.

```
> summary(model)
Call:
glm(formula = Late_delivery_risk ~ ., family = binomial(link = "logit"), data = train)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4768  -0.9789   0.3105   1.2434   1.3963
Coefficients:
                             Estimate Std. Error  z value Pr(>|z|)
(Intercept)                 2.989e+00  3.392e-02   88.114   <2e-16 ***
Product.Card.Id             3.007e-05  2.093e-05    1.437    0.151
Shipping.ModeSame Day      -3.173e+00  3.903e-02  -81.302   <2e-16 ***
Shipping.ModeSecond Class  -1.817e+00  3.465e-02  -52.428   <2e-16 ***
Shipping.ModeStandard Class -3.491e+00  3.242e-02 -107.665   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
   Null deviance: 198848  on 144419 degrees of freedom
Residual deviance: 164437 on 144415 degrees of freedom
AIC: 164447
Number of Fisher Scoring iterations: 5

> confint(model , level=0.95)
2.5 %        97.5 %
(Intercept)                 2.923169e+00  3.056170e+00
Product.Card.Id            -1.095307e-05  7.109202e-05
Shipping.ModeSame Day      -3.250153e+00 -3.097136e+00
Shipping.ModeSecond Class  -1.885168e+00 -1.749310e+00
Shipping.ModeStandard Class -3.555018e+00 -3.427896e+00
```

Figura 70. Montaje y ajuste del modelo de Regresión Logísica.

Ahora se ejecuta la función *anova( )* en el modelo para analizar su desviación. Se observa que al agregar la variable “Shipping Mode”, se reduce significativamente la desviación residual. La variable “Product Card Id” parece mejorar el modelo por su bajo valor de p, como se indica en la Figura 71.

```
> anova(model, test="Chisq")
Analysis of Deviance Table
Model: binomial, link: logit
Response: Late_delivery_risk
Terms added sequentially (first to last)

              Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         144419     198848
Product.Card.Id  1        1    144418     198847    0.396
Shipping.Mode    3    34411    144415     164437   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> exp(coefficients(model))
                (Intercept)            Product.Card.Id             Shipping.ModeSame Day
                19.86888056                1.00003007                0.04186724
  Shipping.ModeSecond Class             Shipping.ModeStandard Class
                 0.16255087                0.03047338
```

Figura 71. Análisis de la Varianza del modelo de Regresión Logísica.

El *odd* es la probabilidad de que suceda un evento dividido por la probabilidad de que no suceda; oscila entre 0 e infinito y se puede calcular para la ocurrencia del evento como para la no ocurrencia del evento [82]. Al analizarlo, se observa que los coeficientes correspondientes a las categorías de la variable “Shipping Mode” son menores que 1, por lo cual se calcula su inversa para poder comparar , sus valores se describen en la Tabla 21.

Shipping.ModeSame Day	23.885
Shipping.ModeStandard Class	32.815
Shipping.ModeSecond Class	6.1519

Tabla 21. Cálculo de *odds* correspondiente a cada categoría de envío.

Se concluye que 23 unidades de envió en el mismo día, aumenta un 88% las posibilidades de que se dé un envió tardío. Para 32 unidades de envió en una Clase Estándar, existe un incremento del 81% de posibilidad de que exista un retraso en el envío. El aumento de 6 unidades de envió de Segunda Clase implica que probabilidad de retraso en el envío sea del 6.5%.

5.3.3. Evaluación de la habilidad predictiva del modelo

La división del conjunto de datos se realiza en dos partes: conjunto de entrenamiento y prueba. El conjunto de entrenamiento se utiliza para ajustar al modelo que posteriormente se ejecuta en el conjunto de prueba.

```
train <- dataset[1:144420,]
test <- dataset[144421:180519,]
```

Se evalúa la funcionalidad del modelo al predecir "y" que tiene los valores de la variable "Late_delivery_risk*"* en un nuevo conjunto de datos. El límite de decisión es 0.5.
Si P (y = 1 | X)> 0.5 entonces y = 1 de lo contrario y = 0.

```
"Accuracy 0.700000000000001"
```

Como último paso, se traza la curva *ROC* como se muestra en la Figura 72 y se calcula el *AUC* (área debajo de la curva) para medir el rendimiento del clasificador binario. La precisión del modelo en la predicción realizada en el conjunto de prueba fue del 70%.

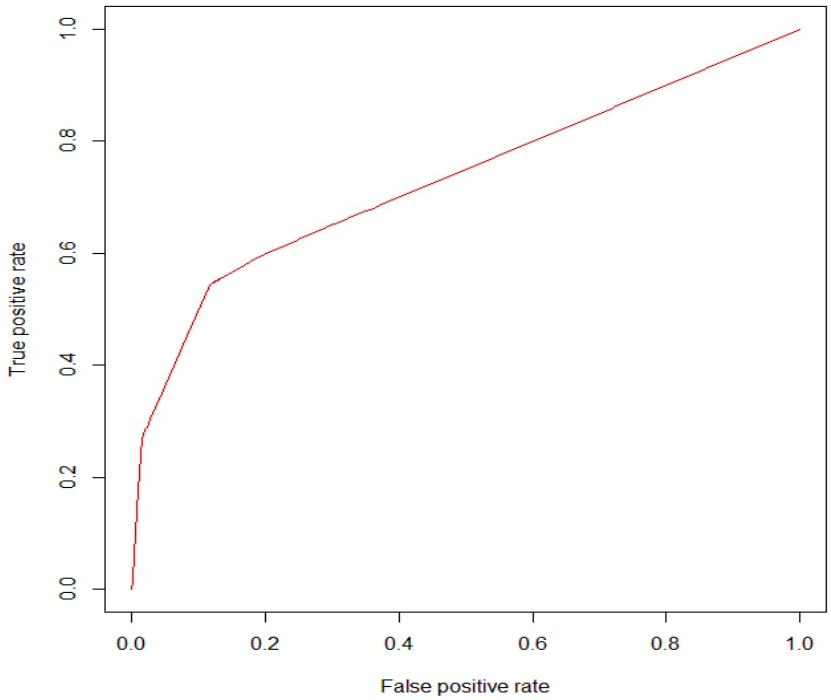


Figura 72. Curva ROC de rendimiento del clasificador binario

Un modelo con buena capacidad predictiva debe tener un AUC más cercano a 1 [81]. En el modelo evaluado da como resultado: `AUC =0.7307086.` Por tanto, se concluye que el modelo es un poco bueno para la predicción.

5.4. Análisis de la Demanda con Regresión Lineal Múltiple

El Método de Mínimos Cuadrados, o Regresión Lineal, se utiliza tanto para pronósticos de series de tiempo como para pronósticos de relaciones causales. En particular cuando la variable dependiente cambia como resultado del tiempo [83].

En el siguiente análisis se desarrolla un pronóstico de la Demanda haciendo uso de información histórica de ventas mensuales de productos correspondientes desde Enero del año 2015, hasta Enero del 2018. Para el análisis se utiliza Lenguaje R, seleccionando las siguientes variables, como se muestra en la Tabla 22. El código fuente del programa se adjunta en el Anexo H.

Nombre de la variable	Tipo	Descripción
Month	<int>	Variable numérica que indica el número de mes
Sales	<dbl>	Ventas mensuales correspondientes a cada año
Quantity	<int>	Cantidad de artículos vendidos en cada mes
Year	<int>	Años correspondiente al mes de venta registrada

Tabla 22. Selección de variables predictoras de Demanda.

5.4.1. Análisis de relación entre variables

Esta información es crítica a la hora de identificar cuáles pueden ser los mejores predictores para el modelo. En la Figura 73, se representa la distribución de cada variable mediante histogramas.

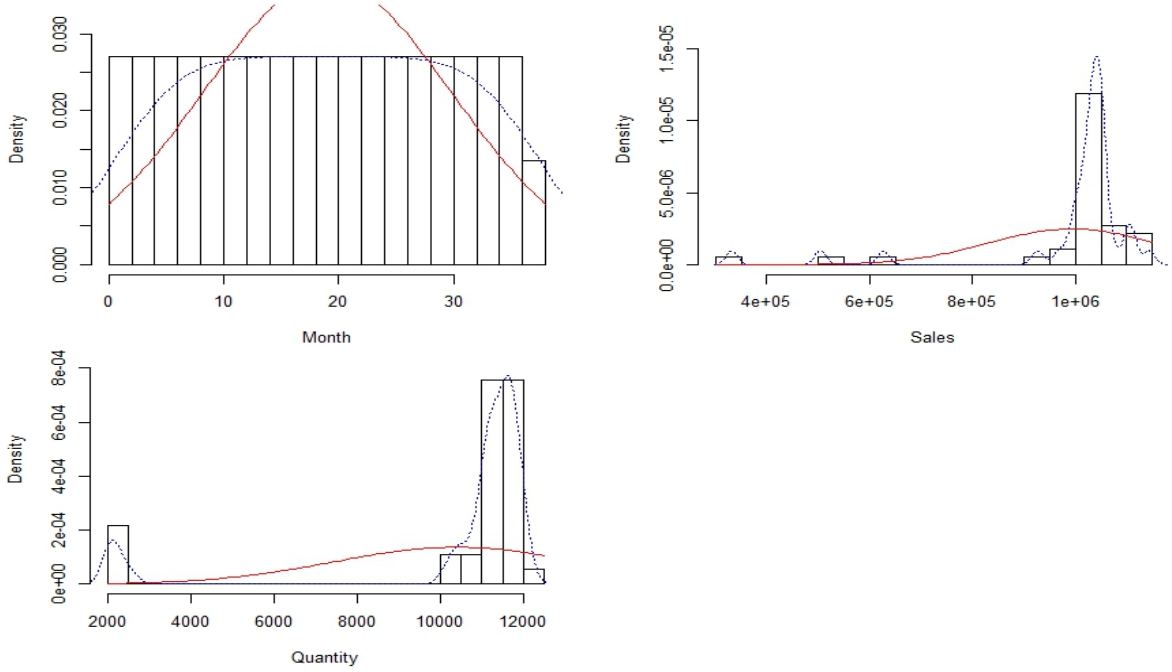


Figura 73. Distribución de variables mediante histogramas.

La matriz de correlación es presentada en la Tabla 23 ,para cada par de variables, indicando que existe una fuerte relación positiva entre las variables “Sales” y “Quantity”, entre “Month” y “Quantity” presenta una correlación negativa por tanto a medida que aumenta el mes la cantidad disminuye.

	Month	**Sales**	**Quantity**
Month	1.000	-0.348	-0.578
Sales	-0.348	1.000	0.795
Quantity	-0.578	0.795	1.000

Tabla 23. Matriz de Corelación.

5.4.2. Montaje y ajuste del modelo

El modelo es ajustado con las variables presentadas en la matriz de correlación, se visualiza en la Figura 74; presenta un intervalo de confianza con un nivel del 95%, para cada uno de los coeficientes parciales de correlación.

```
Call:
lm(formula = Sales ~ ., data = trainingData)

Residuals:
   Min     1Q Median     3Q    Max
-18755  -7414   1734   6872  20756

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.759e+05  6.513e+04   2.700   0.0125 *
Month       2.752e+02  2.661e+02   1.034   0.3113
Quantity    7.370e+01  5.665e+00  13.011  2.3e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10770 on 24 degrees of freedom
Multiple R-squared:  0.877,  Adjusted R-squared:  0.8667
F-statistic: 85.53 on 2 and 24 DF, p-value: 1.204e-11

> confint(lmMod , level=0.95)
                   2.5 %       97.5 %
(Intercept) 41449.81560 310310.16401
Month        -273.92972    824.34959
Quantity       62.01336     85.39703
```

Figura 74. Montaje y ajuste del modelo de Regresión Linel Múltiple.

Por tanto, el modelo predictor queda de la siguiente manera

Y = 175879.98981 + 275.20993*Month + 73.70519*Quantity

Por cada mes en aumento, las ventas aumentan en 175879.99, manteniéndose constantes el resto de predictores.

La Relación lineal entre los predictores numéricos y la variable respuesta son validados mediante diagramas de dispersión. Si la relación es lineal, los residuos deben de distribuirse aleatoriamente en torno a 0 con una variabilidad constante a lo largo del eje X, permitiendo identificar posibles datos atípicos como se indica en la Figura 75. Se concluye que cumple la linealidad para todos los predictores.

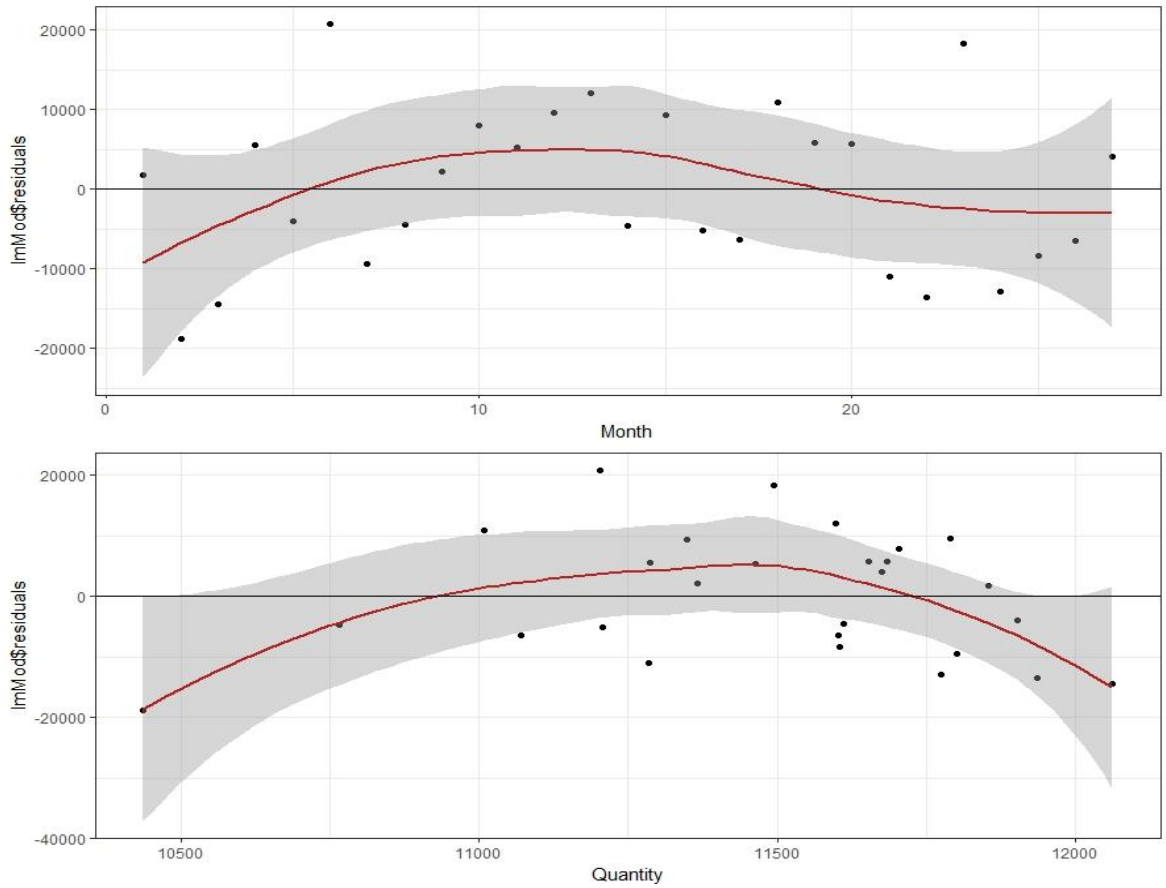


Figura 75. Diagramas de dispersión entre predictores y residuos del modelo.

A continuación, se muestra el test de Normalidad y se visualiza la Distribución Normal de los residuos en la Figura 76.

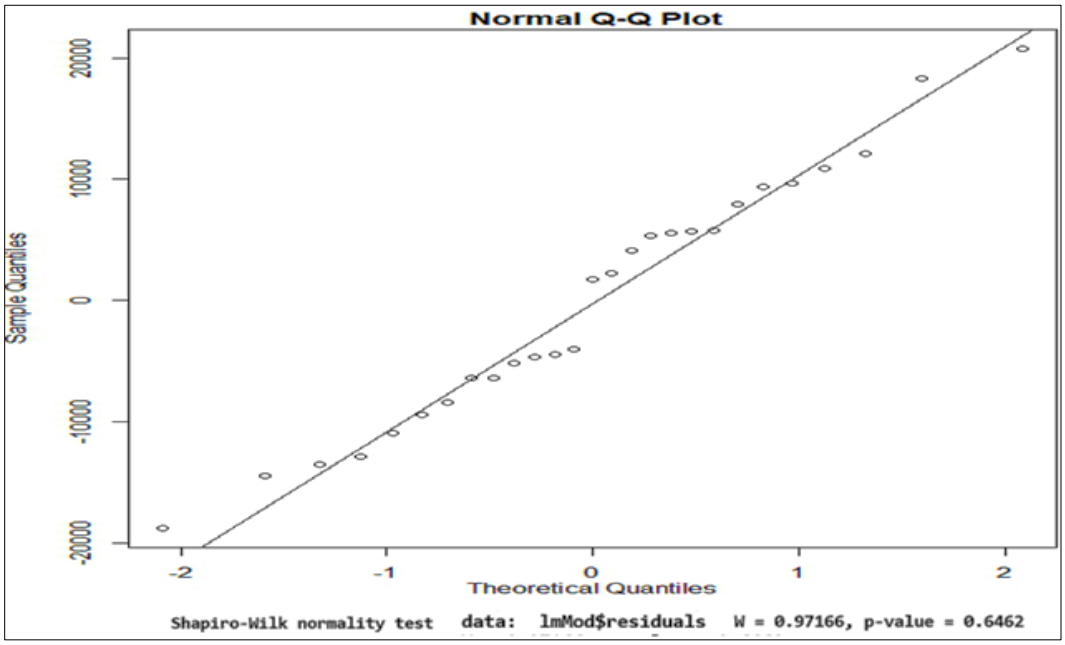


Figura 76. Distribución normal de los residuos.

El análisis gráfico de la Figura 76 y el test de hipótesis confirman la normalidad.

5.4.3. Evaluación de la habilidad predictiva del modelo

La división del conjunto de datos se realiza en dos partes: conjunto de entrenamiento y prueba. El conjunto de entrenamiento es utilizado para ajustar al modelo.

```
trainingData<- data[1:27,]
testData <- data[27:37,]
```

Al calcular las medidas de precisión (accuracy) y tasas de error MAPE (Error Porcentual Absoluto Medio) un indicador del desempeño del pronóstico de Demanda que expresa la exactitud como un porcentaje del error, mientras más bajo es este valor el modelo es mejor. Se averigua la precisión de predicción del modelo [83]. Los valores de la Demanda predecida se registran el la Tabla 23.

Nº de mes	Valores actuales	Valores predecidos
27	1048004.8	1043892.5
28	1038321.6	1008273.3
29	1105485.3	997050.5
30	1032086.5	935487
31	1104373.4	1001875.8
32	1109337.2	1002445.8
33	1143775.1	959013.8
34	1073994.2	368763.1
35	626914.4	336976.5
36	503910.8	342337.4
37	331650.1	342538.9

Tabla 23. Demanda real vs. Demanda predecida.

Las correlaciones entre los valores reales y predecidos son utilizados como una forma de medida de precisión, como se observa en la Figura 77. Una mayor precisión de correlación implica que si los valores actuales aumentan, los pronosticados también aumentan y viceversa [83].

```
> correlation_accuracy
           actuals predicteds
actuals    1.00000    0.79624
predicteds 0.79624    1.00000

> mape
[1] 0.1861681
```

Figura 77. Correlación de valores reales y predecidos

La precisión en la predicción del modelo aplicado al pronóstico es del 79,62 % y en promedio, el pronóstico está errado en un 18.62%. Por tanto, se concluye que el modelo es bueno para la predicción de la Demanda.

5.5. Análisis de Big Data con Tableau

Esta sección analiza los resultados del manejo de transacciones implicadas en la Cadena de Suministro.

5.5.1. Tendencia y pronóstico en ventas

Se analiza los sucesos generados en ventas pasadas y se establece un pronóstico para el 2018, mediante el *dashboard* presentado en la Figura 78 , presentando el movimiento de ventas de los 3 últimos años y el pronóstico obtenido del modelo de regresión lineal múltiple.

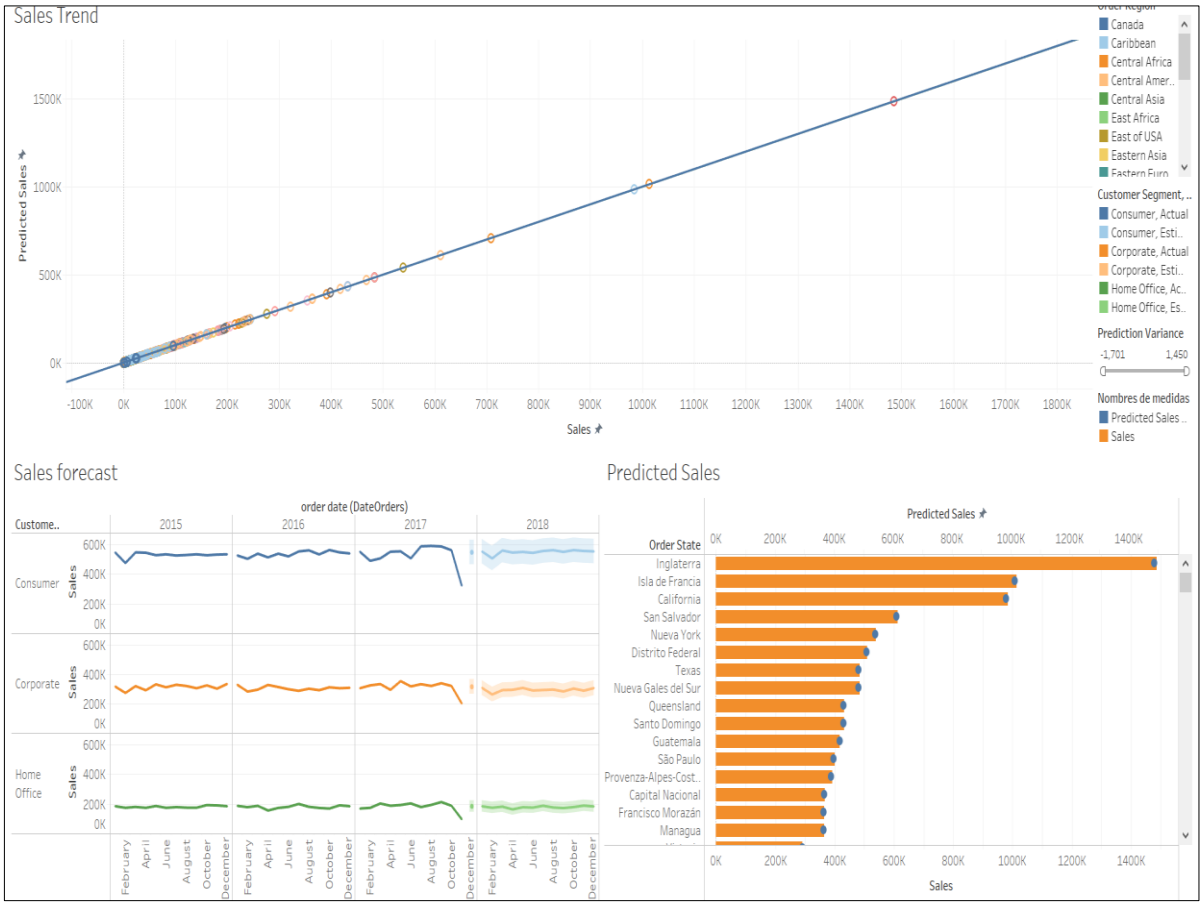


Figura 78. *Dashboard* de tendencia de ventas.

La Tabla 24, compara estimaciones de ventas frete a ventas reales de los 3 años anteriores.

Segmento	Venta real			Pronostico
	2015	2016	2017	2018
Consumer	6,374,924.37	6,453,409.22	6,104,626.32	4,364,304.26
Corporate	3,778,905.37	3,671,523.60	3,613,247.68	2,506,119.00
Home Office	2,187,001.68	2,178,884.50	2,090,562.15	1,322,750.27

Tabla 24. Ventas pronosticadas por segmento de cliente.

Las regiones en las que más se aleja la predicción frente a ventas reales es Oceanía con una variación de 2389.19 disminuyendo la ganancia planificada, mientras que en Asia del Sur se aleja con una variación de $-1431.51, generando pérdidas en función a lo pronosticado. El pronóstico de oferta anual de productos indica que el monto en compra correspondiente al segento *Consumer* para el 2018 es más bajo que en años anteriores, debido a pérdidas registradas en últimos meses. Estas predicciones se basan en un nivel de confianza del 95%.

5.5.2. Clústeres en ventas y beneficios por cliente

Tableau usa el algoritmo *k-means* para *Clustering*, permite especificar un número deseado de clústeres o a su vez sugiere un número óptimo de clústeres. Utiliza el algoritmo de Lloyd con distancias euclídeas cuadradas para calcular el agrupamiento *k-means* para cada *k* [84].

El algoritmo de Lloyd comienza seleccionando centros de clúster iniciales; divide las marcas asignando cada una a su centro más cercano; refina los resultados al calcular nuevos centros para cada partición al promediar todos los puntos asignados al mismo clúster; revisa la asignación de marcas a los clústeres y reasigna las marcas que están ahora más cerca de un centro diferente que antes. Por último, los clústeres se redefinen y las marcas se reasignan de forma iterativa hasta que no se producen más cambios [84].

Para determinar la cantidad optima de conglomerados y evaluar la calidad del clúster, Tableau utiliza el criterio de Calinski-Harabasz, el cual se define como: $\frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$, donde SS_B es la varianza global entre clústeres, SS_W la varianza global dentro del clúster, k el número de clústeres y N el número de observaciones [84].

La Tabla 25, muestra la suma de cuadrados entre grupos, que indica la separación entre los clústeres con un valor de 3.0492. La suma de cuadrados dentro de grupos presenta un valor muy bajo de 0.012045, por tanto, indica que existe una buena cohesión entre los conglomerados. La varianza explicada del modelo es de $0.996047 \approx 1$, por tanto, se concluye que es un buen modelo.

Entradas para la agrupación en clústeres		**Resumen de diagnósticos**	
Variables:	Suma de Order Item Profit Ratio	**Número de clústeres:**	6
	Suma de Order Item Total	**Número de puntos:**	14033
	Suma de Order Profit Per Order	**Suma de cuadrados entre grupos:**	3.0492
Nivel de detalle:	Customer Fname, Customer Lname	**Suma de cuadrados dentro de grupos:**	0.012045
Escala:	Normalizada	**Suma de cuadrados total:**	3.0613

Tabla 25. Resultados del modelo de clúster aplicado.

En la Figura 79, se presenta el *dashboard* de ventas y beneficios por cliente, en el cual se realiza un análisis de Clúster *k – means* .

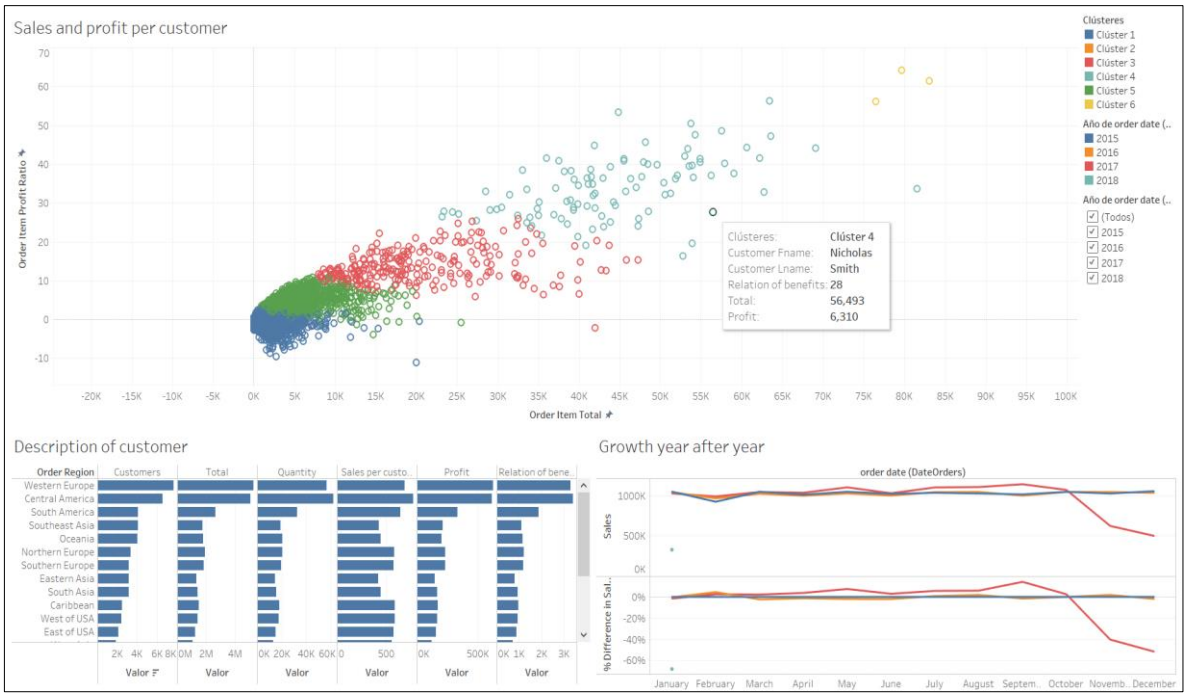


Figura 79. *Dashboard* de análisis de Clúster en ventas y beneficios por cliente

La Tabla 26, presenta información de los centros de cada clúster. El clúster 1 contiene el mayor grupo de clientes e indica el menor porcentaje de beneficio que es de 13,5%, por un monto total en ordenes de $ 744.14, teniendo a su vez el menor beneficio por orden de $24,81. El clúster 2 es el más pequeño con solo un cliente que presenta el mayor porcentaje de beneficio, mayor monto en orden de $522800,00, siendo el más beneficioso y rentable.

Centros				
Clústeres	**Número de elementos**	**Suma de *Order Item Profit Ratio***	**Suma de *Order Item Total***	**Suma de *Order Profit Per Order***
Clúster 1	11107	0.1349	744.14	24.813
Clúster 2	1	2937.6	4286000	522800
Clúster 3	258	13.486	19737	2395.1
Clúster 4	99	33.582	44784	6040.9
Clúster 5	2560	3.8629	3954.2	716.62
Clúster 6	8	81.239	106840	14747
Sin clústeres	0			

Tabla 26. Información de centros de clústeres.

5.5.3. Análisis del estado actual de ventas

En este apartado presenta un análisis de la situación actual de ventas registradas a nivel global. La Figura 80, muestra un *dashboard* en el que se observan las ventas globales y sus respectivos beneficios. Los clientes adquieren sus productos en mayor cantidad en EEUU, Europa y Asia Central.

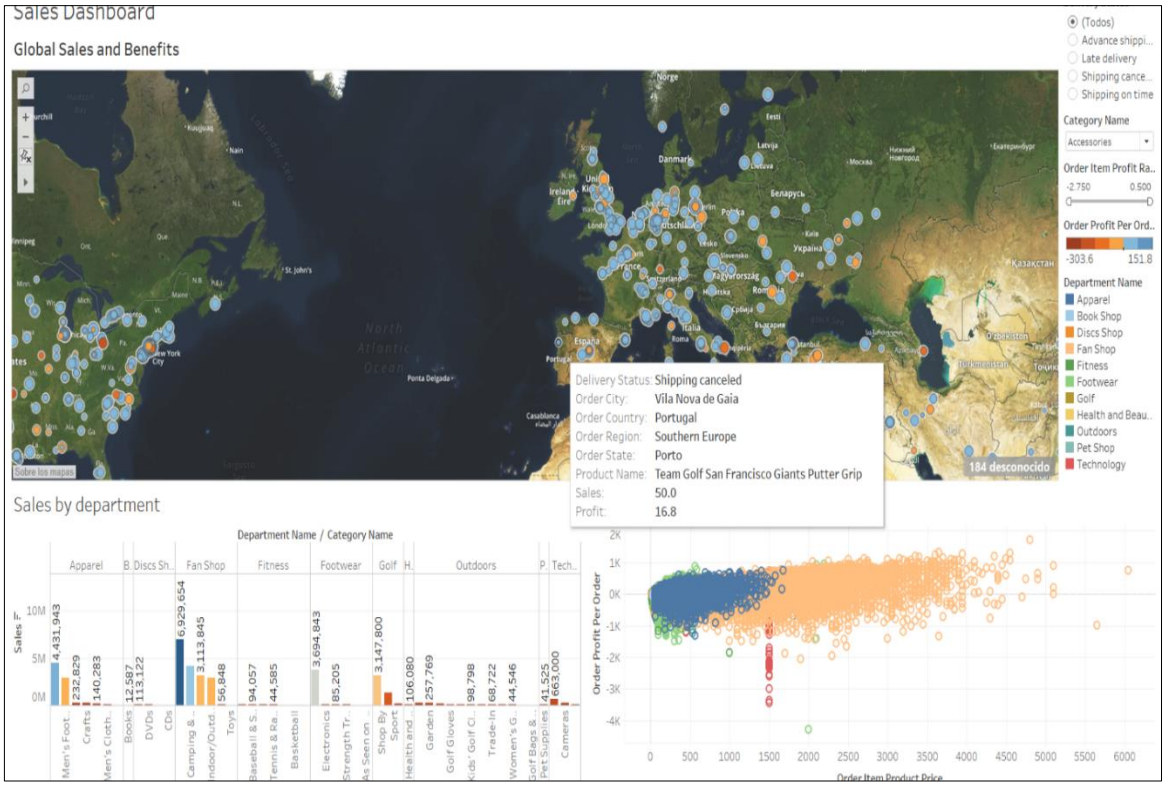


Figura 80. Dashboard de ventas.

La Tabla 27 , indica los valores de ventas globales y beneficios según el estado de envío.

Estado de envío	Región	Ventas Mínimas		Región	Ventas Máximas
Shipping on time	Asia Central	$	23,215.23	*Central America*	$ 1,017,388.88
Advance shipping	Asia Central	$	24,111.34	*Western Europe*	$ 1,330,235.80
Shipping canceled	Asia Central	$	2,779.66	*Western Europe*	$ 265,442.21
Late Delivery	Asia Central	$	59,733.70	*Western Europe*	$ 3,292,013.64

Tabla 27. Análisis de ventas por Región y estado de envió.

Asia Central presenta ventas mínimas en todos los estados de envío; mientras que en *Western Europe* es la región que registra la mayor cantidad de venta generadas en envíos tardíos. La Tabla 28, registra ventas globales por departamento.

Departamento	Categoría	Ventas Mínimas	Categoría	Ventas Máximas
Apparel	*Baby*	$ 12,229.56	*Cleats*	$ 4,431,942.78
Book Shop	*Book Shop*	$ 12,587.40		
Discs Shop	*CDs*	$ 3,059.59	*Music*	$ 113,122.10
Fan Shop	*Toys*	$ 6,104.66	*Fishing*	$ 6,929,653.69
Fitness	*Soccer*	$ 26,477.05	*Sporting Goods*	$ 117,006.75
Footwear	*As Seen on TV!*	$ 20,597.94	*Cardio Equipment*	$ 3,694,843.20
Golf	*Girls' Apparel*	$ 151,706.20	*Women's Apparel*	$ 3,147,800.00
Health and Beauty	*Health and Beauty*	$ 106,080.48		
Outdoors	*Golf Bags & Carts*	$ 10,369.39	*Garden*	$ 257,768.73
Pet Shop	*Pet Supplies*	$ 41,524.80		
Technology	*Consumer Electronics*	$ 108,991.28	*Computers*	$ 663,000.00

Tabla 28. Análisis de ventas por departamento y categoría.

La categoría *CDs* del departamento *Discs Shops* presenta la menor cantidad de ventas y la máxima venta en dólares es obtenida por de la categoría *Fishing* del departamento *Fan Shop*. La Tabla 29 indica el desglose de clientes con mayor beneficio y perdida por departamento.

Departamento	**Genera mayor beneficio**			**Genera mayor perdida**		
	Id cliente	**Nombre**	**Beneficio**	**Id cliente**	**Nombre**	**Pérdida**
Apparel	11048	Randy Ball	$ 744.98	2610	Virginia Johnson	$ (1,307.66)
Book Shop	17595	Stella Blanchard	$ 15.23	12468	Paloma Mitchell	$ (76.15)
Discs Shop	16116	Ivana Walters	$ 130.33	16208	Cassady Chambers	$ (552.58)
Fan Shop	2641	Betty Spears	$ 1,708.96	9135	Mary Johnson	$ (2,044.40)
Fitness	4844	Lauren Wu	$ 465.50	4077	Amy Smith	$ (1,844.98)
Footwear	5533	Mary Harrison	$ 949.99	1428	Mary Clark	$ (4,274.98)
Golf	6448	Mary Lawrence	$ 518.59	12180	Mary Allen	$ (1,076.09)
Health and Beauty	15718	Cassandra Landry	$ 145.05	15710	Rosalyn Chandler	$ (685.71)
Outdoors	11015	Mary Martinez	$ 452.90	15419	Iona Noel	$ (1,222.27)
Pet Shop	16467	Zenaida Johnson	$ 41.36	19052	Lacy Lopez	$ (208.89)
Technology	14111	Molly Gilliam	$ 720.30	14086	Nelle Hyde	$ (3,442.50)

Tabla 29. Desglose de clientes.

El departamento que más beneficio ha generado es *Fan Shop* por Betty Spears. Sin embargo, la mayor pérdida es generada en el departamento de *Footwear*, por Mary Clark.

5.5.4. Análisis de compras y tendencias de envíos

Esta sección presenta un análisis de compras registradas de acuerdo al estatus de entrega y a la rentabilidad de órdenes. La Figura 81, muestra un *dashboard* en el que se observan las compras y beneficios por clientes.

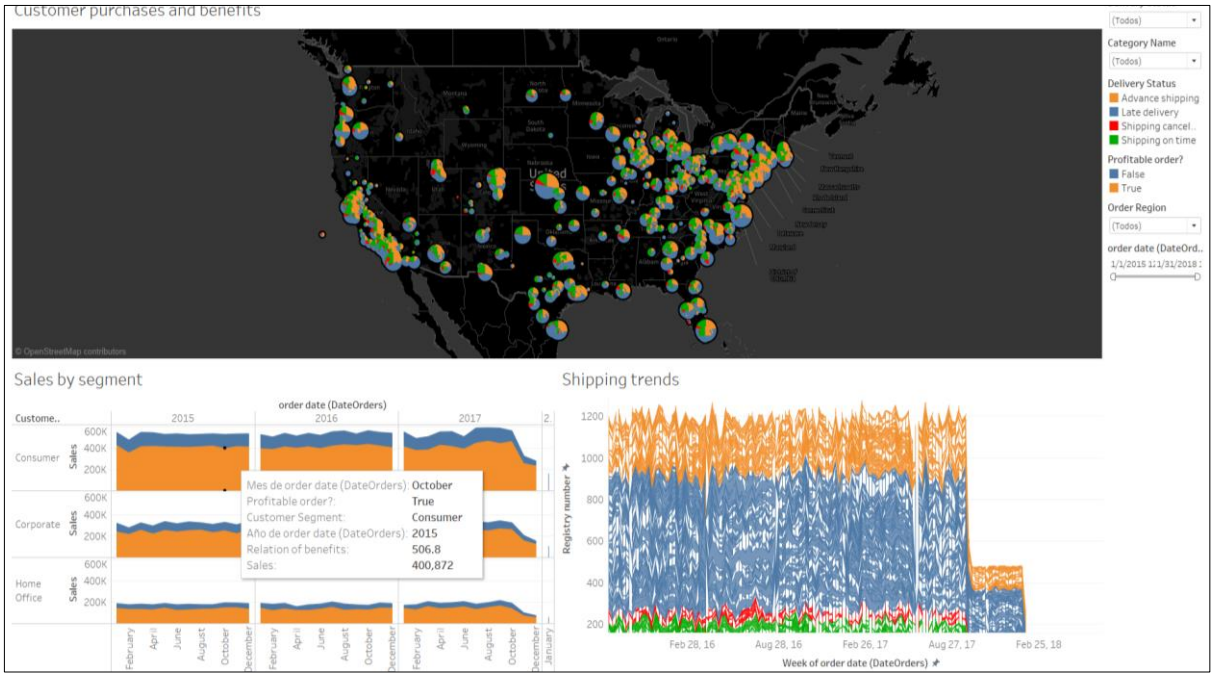


Figura 81. *Dashboard* de compras de clientes y beneficios.

La Tabla 30, indica las ventas de ordenes generadas por segmento de cliente.

Segmento de cliente	Ventas	Estado	Ventas por estado en $	% representativa
Consumer	$ 19,095,790.16	PR	7292280.76	38%
Corporate	$ 11,168,406.84	PR	4245835.11	38%
Home Office	$ 6,520,538.02	PR	2612125.90	40%

Tabla 30. Ventas generadas por Segmento de cliente

El segmento *Consumer* global genera $19'095790.16, del cual el estado de Puerto Rico genera $ 7'292280.76, representando el 38% del total de las ventas de dicho segmento. La Tabla 31, presenta al cliente que mayormente ha participado en cada segmento. El cliente Mary Smith, del estado de Texas, del segmento *Consumer*, es la cliente potencial al cual no se puede descuidar ya que genera ventas totales de $5 599.72.

Análisis de ventas por segmento y por cliente					
Segmento	**Id Cliente**	**Nombre**	**ESTADO**	**CIUDAD**	**Ventas**
Consumer	4798	Mary Smith	TX	PLANO	$ 5,599.72
Corporate	291	John Smith	CT	BRISTOL	$ 4,799.76
Home Office	664	Booby Jimenes	MI	HOLLAND	$ 3,999.80

Tabla 31. Análisis de venta por segmento y por cliente

La Tabla 32, indica el total de las ventas globales anuales por segmento de cliente.

Segmento de Cliente	Ventas Globales					
	Ordenes rentables			Ordenes no rentables		
	2015	2016	2017	2015	2016	2017
Consumer	$4,965,879.35	$5,010,927.66	$4,783,788.30	$1,409,045.03	$1,442,481.56	$1,320,838.02
Corporate	$2,934,455.51	$2,846,606.91	$2,850,767.26	$844,449.87	$824,916.70	$762,480.41
Home Office	$1,693,778.29	$1,670,101.31	$1,647,440.55	$493,223.39	$508,783.19	$443,121.60
Total	$9,594,113.15	$9,527,635.88	$9,281,996.11	$2,746,718.28	$2,776,181.45	$2,526,440.04

Tabla 32. Ventas por año por segmento

Se verifica que el segmento *Home Office* registra menor valor de ventas y en el transcurso de los años ha ido reduciendo; mientras que *Consumer* es el segmento que más ventas registra, demostrando que va en crecimiento este segmento.

5.5.5. Análisis de ventas y beneficios por mercado

En este apartado se realiza el análisis global en cada mercado, como se observa en la Tabla 33.

Mercado	Beneficio por orden	Ventas
Africa	$ 252,071.18	$ 2,294,452.93
Europe	$ 1,169,442.96	$ 10,872,396.80
LATAM	$ 1,123,321.61	$ 10,277,612.84
Pacific Asia	$ 857,753.44	$ 8,273,743.74
USCA	$ 564,313.78	$ 5,066,528.71

Tabla 33. Ventas y Utilidades por Mercado

La Figura 82, muestra un *dashboard* que representa el desenvolvimiento de las ventas por mercado. Se observa que Europa genera mayores ventas obteniendo utilidades de $1’169.442,96. África registra ventas muy bajas presentando utilidades que incentiva a planificar el estudio de mercado y mejorar sus ventas.

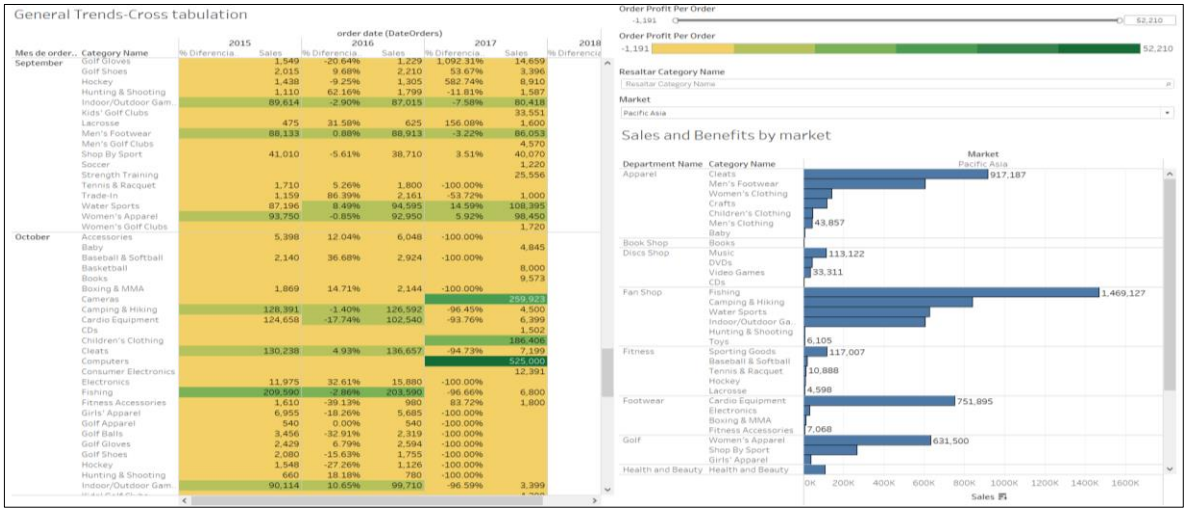


Figura 82. *Dashboard de* ventas y beneficios por mercado.

La Tabla 34, muestra beneficios y perdidas en ventas totales por mercado, departamento y categoría. LATAM tiene el mayor beneficio en el departamento *Fan Shop* de la categoría *Fishing*; mayor valor en pérdidas se registra en la categoría *As Seen on TV*, del departamento *Footwear*.

Menor Beneficio				
Mercado	**Departamento**	**Categoría**	**Beneficio**	**Ventas**
Africa	Footwear	Boxing & MMA	$ (151.05)	$ 5,222.15
Europe	Discs Shop	CDs	$ 193.40	$ 1,501.57
LATAM	Footwear	As Seen on TV!	$ (398.52)	$ 7,599.24
Pacific Asia	Book Shop	Books	$ 32.23	$ 3,014.76
USCA	Fitness	Lacrosse	$ 331.96	$ 3,448.62
Mayor Beneficio				
Mercado	**Departamento**	**Categoría**	**Beneficio**	**Ventas**
Africa	Fan Shop	Fishing	$ 48,234.46	$ 473,976.31
Europe	Fan Shop	Fishing	$ 207,658.61	$ 1,925,903.75
LATAM	Fan Shop	Fishing	$ 223,265.75	$ 2,033,498.38
Pacific Asia	Fan Shop	Fishing	$ 148,673.22	$ 1,469,126.58
USCA	Fan Shop	Fishing	$ 128,388.72	$ 1,027,148.67

Tabla 34. Beneficios en ventas totales por mercado.

5.5.6. Análisis *Clickstream* de productos visitados

Los informes de Clickstream ayudan a analizar e impulsar la generación de ingresos, identificando las causas de los porcentajes de rebote, como el abandono del carrito y proporcionar recomendaciones de productos basadas en usos de múltiples dispositivos [70].

La Figura 83, presenta un *dashboard* en elque se realiza el análisis de Clickstream de los productos visitados.

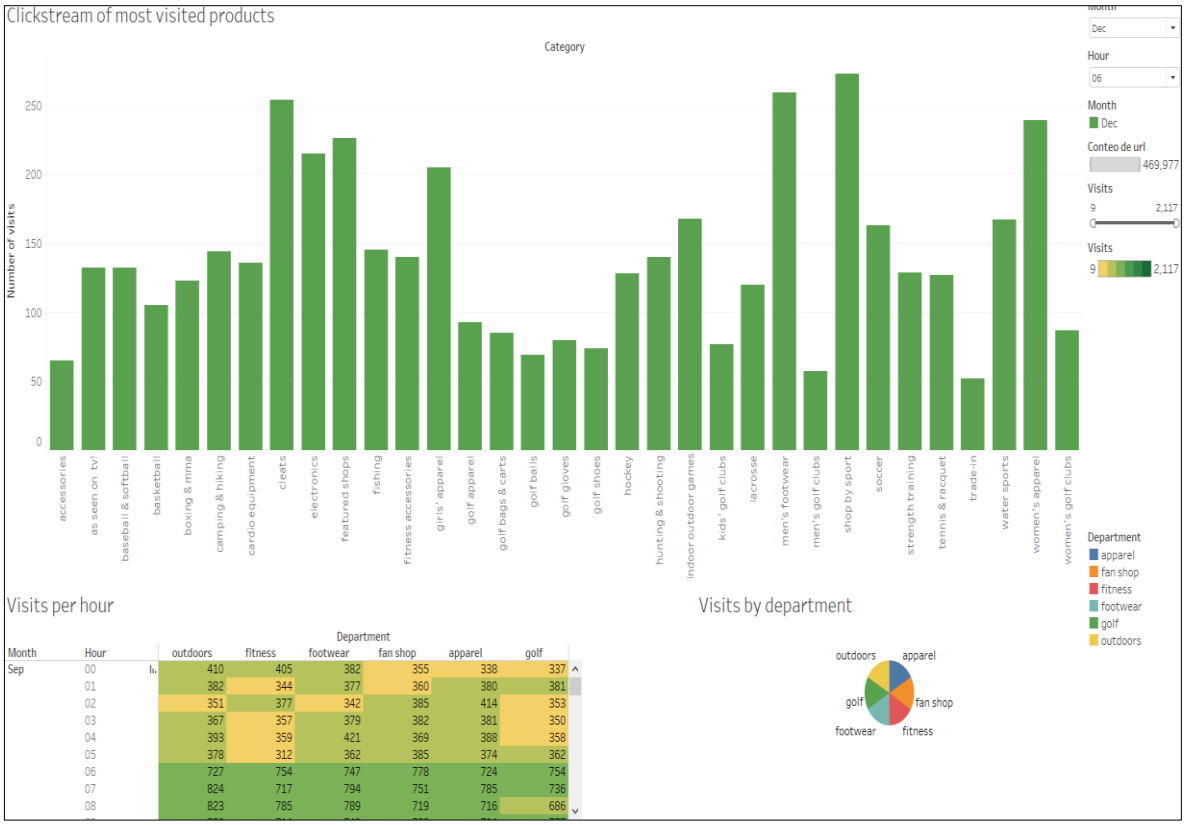


Figura 83. *Dashboard* de *Clikstream* de productos visitados.

En la Tabla 35, se observa que el departamento *"outdoors"* es el más visitado y *"golf"* el menos visitado. En la Tabla 36 ,se muestra que a las 20 horas es cuando más visitas se registraron en el mes de Septiembre y los departamentos *"golf"* y *"fitness"* presentaron la menor cantidad de visitas en los meses de Octubre y Diciembre. La Tabla 37, indica que el total de visitas en el mes de Septiembre fue el más alto, Noviembre fue el mes más bajo.

Departamento	Visitas
apparel	79319
fan shop	78724
fitness	76437
footwear	79136
golf	76435
outdoors	79926

Tabla 35. Visitas por departamento

Departamento	Horas	Mes	Visitas
apparel	20 h	Sep	2117
golf	5 h	Oct	9
fitness	5 h	Oct	9
golf	5 h	Dec	9
fitness	5 h	Dec	9

Tabla 36. Visitas por hora

Mes	Visitas
Sep	137238
Oct	84205
Nov	80860
Dec	84093
Jan	83581

Tabla 37. Visitas por mes

La Tabla 38, registra la categoría de producto más visitada, la hora y mes en donde presenta la mayor cantidad de visitas, siendo este, a su vez, uno de los productos favoritos a comprar.

Categoría	featured shops
Producto	adidas Kids' RG III Mid Football Cleat
Departamento	apparel
Hora	20 h
Mes	Sep
Visitas	744
Url	/department/apparel/category/featured%20shops/product/ adidas%20Kids'%20RG%20III%20Mid%20Football%20Cleat

Tabla 38. Categoría y producto más visitado.

5.6. Síntesis

En este capítulo se implementó modelos de *Machine Learning* y Minería de Datos en Lenguaje R, para la detección y predicción de fraude, análisis de Cesta de Mercado, pronósticos de envíos tardíos y análisis de la Demanda mensual.

Una de las aplicaciones más importantes de *Machine Learning* para BDA en Cadenas de Suministro fue la detección de fraude. Basado en el tipo de variable de resultado, se utilizó cuatro modelos diferentes determinando cuál se ajustó mejor al conjunto de pruebas en función de la curva ROC y la precisión. Cada modelo se probó contra 100.000 transacciones aleatorias, capturando el tiempo de procesamiento y estimando la tasa de falsos positivos, dando como resultado el mejor modelo a utilizar Random *Forest.*

Con la ayuda de Tableau y R se realizaron *Dashboards*, que permitieron realizar un análisis visual en tiempo real al conectarse con los datos alojados en Hive (dentro de la plataforma CDH). Los *Dashboards* interactivos ayudaron a descubrir información oculta al instante.

Hadoop acelero BDA en Cadenas de Suministro, a través de procesos distribuidos, por lo tanto, proporciono respuestas rápidamente. La extensibilidad y simplicidad del marco son los diferenciadores clave que lo convierten en una herramienta prometedora para el procesamiento de datos.

El cambio mejorado de datos a través de la transparencia y rendimiento aumenta las segmentaciones del mercado, brindando mayor soporte a la toma de decisiones a través de los análisis avanzados y mayor capacidad para innovar servicios y modelos de negocio. Las compañías deben seguir las tendencias en *Big Data* cuidadosamente para tomar las decisiones que mejor se adapten a sus negocios.

Esta página fue intencionalmente dejada en blanco

6. Conclusiones

Con el aumento de los dispositivos y sensores inteligentes, la producción de datos ha aumentado en los últimos años. La interacción entre *Big Data* e IoT se encuentra actualmente en sus primeras etapas de desarrollo, donde es necesario procesar, transformar y analizar grandes cantidades de datos con alta frecuencia.

Small Data proporciona información sobre cómo un proceso en particular funciona un momento dado. Ahora para encontrar respuestas a 'por qué' está funcionando de esa manera, se necesita de *Big Data*; entonces se complementa el uno al otro, haciéndolos susceptibles al BDA.

Big Data significa también grandes sistemas, grandes desafíos y grandes ganancias, por lo que son necesarios más trabajos de investigación en estos subcampos para resolverlo. Afortunadamente se está presenciando el nacimiento y desarrollo de *Big Data*, y ninguna persona puede resolverlo solo. Los recursos humanos, las inversiones de capital y las ideas creativas son componentes fundamentales de su desarrollo.

Esta investigación fue dirigida con el objetivo de proponer un marco que permita la explicación y análisis de datos provenientes de IoT. Se realizó un estudio de la de la influencia y relación existente entre BDA y *Small Data* en IoT, también se discutieron tipos, métodos y tecnologías analíticas de Big Data para la Minería de datos de gran tamaño. Fueron presentados casos de uso notables, analizando varias oportunidades proporcionadas por el análisis de datos en el paradigma IoT, sus desafíos en investigaciones abiertas como la Privacidad, la Minería de datos, la Visualización de datos y la Integración.

La Cadena de Suministro es un medio de comunicación por lo que el uso de tecnologías con sensores IoT, Big Data aporta avances de vanguardia y nuevas oportunidades de negocio relacionadas con la experiencia del cliente a través de todos los canales de venta y dispositivos. El análisis de sus datos permite la generación de conocimiento y toma decisiones más acertadas en las organizaciones.

La disertación fue aplicada, al caso de uso específico “Cadenas de Suministro Inteligente”, presentando, el análisis de los datos generados desde una “Plataforma de Compra-Venta y Control de Stocks de Productos”, desde tecnologías RFID y NFC, para la gestión en tiempo real de la rotación de inventarios y transacciones relacionados con el manejo de Suministro, con el fin de proponer un protocolo de obtención de conocimiento en base al BDA.

La implementación de la arquitectura propuesta permitió la generación, extracción e ingestión de datos provenientes de IoT y de la “Plataforma de Compra-Venta y Control de Stocks de Productos” en HDFS, su visualización y análisis en tiempo real con modelos de *Machine Learning* y Minería de Datos de datos, generando conocimiento y una buena toma de decisiones.

La solución presento cuatro principales componentes de tecnología: Sensores IoT con tecnología RFID/NFC Modulo RC522, una Base de Datos de gestión de la Cadena de Suministro de una empresa mayorista, *Big Data* no estructurada generada a través de los Sensores de IoT RFID/NFC y, por último, la utilización de herramientas *open source* de BDA. En lugar de manejar sensores físicos, se utilizó un programa Python generador de registros Web, para simular la información dada por el sensor, al realizar alguna transacción o visita a la plataforma Web en tiempo real.

A partir de esta investigación, se entiende que cada plataforma de *Big Data* tiene su enfoque individual; algunas están diseñados para el procesamiento por lotes, mientras que otras son buenas para el análisis en tiempo real. Como ejemplo Hadoop frente a los sistemas tradicionales proporciono una escala y flexibilidad mucho mayores.

La plataforma CDH, permitió el almacenamiento, análisis, visualización y exploración de datos en tiempo real, utilizando Flume, Solr y Morphlines, facilitando el manejo de flujos de trabajo *Big Data* de extremo a extremo. Tableau y R se integraron permitiendo ejecutar algoritmos de *Machine Learning* que al final ayudaron a realizar predicciones con un alto grado de probabilidad y posteriormente adquirir conocimiento en las tomas de decisiones.

6.1. Trabajos Futuros

Expresar los requisitos de acceso a los datos de la aplicación y diseñar abstracciones del lenguaje de programación para explotar el paralelismo es una necesidad inmediata [85].

Cada una de las herramientas tiene sus propias ventajas y limitaciones, el desarrollo de herramientas más eficientes para lidiar con los problemas inherentes a *Big Data*, deben tener una disposición para manejar datos y desequilibrios ruidosos, incertidumbre e inconsistencia y valores perdidos.

Las diferentes técnicas utilizadas para el análisis incluyen análisis estadístico, aprendizaje automático, extracción de datos, análisis inteligente, computación en la nube y procesamiento de flujo de datos. En el futuro, es deseable que los investigadores presten más atención a estas técnicas para resolver problemas de *Big Data* de manera efectiva y eficiente.

Se recomienda como trabajo futuro, realizar el BDA de la Cadena de Suministro utilizando sensores IoT físicos RFID/NFC Modulo RC522 e integrándolos a varios Sistemas de Información, con el fin de poder obtener resultados más auténticos en tiempo real.

El BDA generado desde IoT, es un campo muy amplio de búsqueda que se puede aplicar en otros casos de uso específicos como: Edificios Inteligentes, Medición Inteligente, Transporte Inteligente, Agricultura Inteligente, Redes Inteligentes y Sistemas de Semáforos Inteligentes. Queda en manos de los investigadores a futuro continuar con el trabajo.

Bibliografía

[1] M. Marjani *et al.*, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," *IEEE Access*, vol. PP, no. 99, p. 1, 2017.

[2] B. Purcell, "The emergence of " Big Data " technology and analytics," no. October, 2016.

[3] S. Rowe and M. Pournader, "How big data is shaping the supply chains of tomorrow," *KPMG , Supply Chain Big Data Series*, no. March, Sidney , Australia, pp. 1–16, 2017.

[4] M. Vecchio, "IoT and Big Data: An Extraordinary Synergy," *May 26*, 2017. [Online]. Available: https://www.hindawi.com/journals/wcmc/si/793163/cfp/. [Accessed: 02-Oct-2017].

[5] K. Ashton, "That 'Internet of Things' Thing," *RFID Jornal*, 2009. [Online]. Available: http://www.rfidjournal.com/articles/view?4986. [Accessed: 05-Oct-2017].

[6] K. P. Madjid Tavana, "Handbook of Research on Organizational Transformations through Big Data Analytics," 2012, p. 109.

[7] R. Shumway and M. Harrison, "Finding growth through big data," p. 11, 2012.

[8] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC Anal. Futur.*, 2012.

[9] U. S. Profile, "T H E D I G I T AL U N I V E R S E I N 2 0 2 0 : B i g D a t a , Bigger Digital Shadows , and Biggest Growth in the Far East — United States," pp. 1–7, 2013.

[10] M. Beyer, "Gartner says solving `Big Data' challenge involves more than just managing volumes of data," *AaltoDoc, Aalto UnivAalto Univ.*, 2011.

[11] and M. C. R. Mital, J. Coughlin, "Using big data technologies and analytics to predict sensor anomalies," *Proc. Adv. Maui Opt. Sp. Surveill. Technol. Conf*, p. 84, 2014.

[12] N. Golchha, "Big data-the information revolution," *Int. J. Adv. Res.*, vol. 1, pp. 791–794, 2015.

[13] O. K. and N. B. L. Shin, "Data quality management, data usage experience and acquisition intention of big data analytics," pp. 387–394, 2014.

[14] J. C. Alvarado, "Estudio descriptivo de técnicas aplicadas en herramientas Open Source y comerciales para visualización de ...," no. January 2017, 2016.

[15] I. A. T. Hashem, "The rise of `big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, 2015.

[16] D. P. and K. Ahmed, "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 511–518, 2016.

[17] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," *Data Min. Knowl. Discov.*, vol. 2, no. 3, pp. 283–304, 1998.

[18] T. A. S. Foundation., "Apache Hadoop," 2014. [Online]. Available: http://hadoop.apache.org/. [Accessed: 16-Oct-2017].

[19] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Proc. 6th Symp. Oper. Syst. Des. Implement.*, pp. 137–149, 2004.

[20] S. Kaisler, F. Armour, W. Money, and J. A. Espinosa, "Big Data Issues and Challenges," vol. 5, no. 2013, pp. 2013–2015, 2015.

[21] E. Consulting, “THE IMPORTANCE OF SCALABILITY IN BIG DATA PROCESSING.” [Online]. Available: http://blog.eccellaconsulting.com/the-importance-of-scalability-in-big-data-processing. [Accessed: 18-Oct-2017].

[22] P. Hanrahan, “Tableau,” 2017. [Online]. Available: https://www.tableau.com/es-es/resource/business-intelligence. [Accessed: 13-Nov-2017].

[23] C.-W. Tsai, “Big data analytics: A survey,” vol. 2, pp. 1–32, 2015.

[24] V. Estivill-Castro, “Why so many clustering algorithms: A position paper,” in *ACM SIGKDD Explorations Newslett*, 2002, pp. 65–75.

[25] F. C. et Al, “Data mining for the Internet of Things: Literature review and challenges,” vol. 12.

[26] C. Hu, “Data-driven method based on particle swarm optimization and K-nearest neighbor regression for estimating capacity of lithium-ion battery,” vol. 129, pp. 49–55.

[27] A. M. Pedro Larranaga, Inaki Inza, “Clustering,” pp. 1–11, 2008.

[28] and G. M. L. Atzori, A. Iera, “The Internet of Things: A survey,” 2010, pp. 2787–2805.

[29] H.-C. H. and C.-H. Lai, “Internet of Things architecture based on integrated PLC and 3G communication networks,” *IEEE Access*, pp. 853–856.

[30] I. Journal, A. Science, and E. Technology, “Implementation on Data Cleaning for RFID and,” vol. 5, no. Iii, pp. 408–418, 2017.

[31] X.-Y. Chen and Z.-G. Jin, “Research on Key Technology and Applications for Internet of Things,” *Phys. Procedia*, vol. 33, pp. 561–566, 2012.

[32] G. M. L. A. D. Giusto, A. Iera, “The Internet of Thinngs,” in *The Internet of Thinngs.*, 2010.

[33] C. Sun, “Application of RFID Technology for Logistics on Internet of Things,” *AASRI Procedia*, vol. 1, pp. 106–111, 2012.

[34] B. Karakostas, “A DNS architecture for the internet of things: A case study in transport logistics,” *Procedia Comput. Sci.*, vol. 19, no. Ant, pp. 594–601, 2013.

[35] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future internet: The internet of things architecture, possible applications and key challenges,” *Proc. - 10th Int. Conf. Front. Inf. Technol. FIT 2012*, no. April 2017, pp. 257–260, 2012.

[36] G. Chavira, S. W. Nava, R. Hervás, J. Bravo, and C. Sánchez, “Combining RFID and NFC technologies in an AmI conference scenario,” *Proc. Mex. Int. Conf. Comput. Sci.*, pp. 165–172, 2007.

[37] R. Weinstein, “RFID: A technical overview and its application to the enterprise,” *IT Prof.*, vol. 7, no. 3, pp. 27–33, 2005.

[38] N. Mishra, C. C. Lin, and H. T. Chang, “A Cognitive Adopted Framework for IoT Big-Data Management and Knowledge Discovery Prospective,” *Int. J. Distrib. Sens. Networks*, vol. 2015, no. March, pp. 1–13, 2015.

[39] W. Rittmeyer, “IoT Analytics: Big Data, Small Data and Other Data,” *DZone / IoT Zone*, 2015. [Online]. Available: https://dzone.com/articles/iot-analytics-big-data-small. [Accessed: 17-Nov-2017].

[40] Cognizant Digital Works, “Why Small Data Is the Future of IoT,” *idea couture*, 2017.

[Online]. Available: https://ideacouture.com/iot-trends/why-small-data-is-the-future-of-iot/. [Accessed: 04-Aug-2017].

[41] M. Kavis, "Forget Big Data -- Small Data Is Driving The Internet Of Things," *Forbes*, 2015. [Online]. Available: https://www.forbes.com/sites/mikekavis/2015/02/25/forget-big-data-small-data-is-driving-the-internet-of-things/#f273faf5d7e8. [Accessed: 04-Aug-2018].

[42] A. Banafa, "Small Data vs. Big Data : Back to the basics," *Linkedin*, 2014. [Online]. Available: https://www.linkedin.com/pulse/20140703195144-246665791-small-data-vs-big-data-back-to-the-basics. [Accessed: 20-Nov-2017].

[43] M. Goetschalckx, *Supply Chain Engineering*, Springer. Boston - US., 2011.

[44] and R. J. V. Lummus, Rhonda R, "Defining supply chain management: a historical perspective and practical guidelines," *Ind. Manag. Data Syst.*, pp. 11–17, 1999.

[45] Argenis Bauza, "El Internet de las cosas y la cadena de suministro," *KPMG*, 2016. [Online]. Available: https://home.kpmg.com/mx/es/home/tendencias/2016/12/internet-de-las-cosas-y-cadena-de-suministro.html. [Accessed: 11-Dec-2017].

[46] T. Fukui, "A systems approach to big data technology applied to supply chain," *Proc. - 2016 IEEE Int. Conf. Big Data, Big Data 2016*, pp. 3732–3736, 2017.

[47] M. Radhakrishnan, S. Sen, S. Vigneshwaran, A. Misra, and R. Balan, "IoT+Small Data: Transforming in-store shopping analytics & services," *2016 8th Int. Conf. Commun. Syst. Networks, COMSNETS 2016*, no. January, 2016.

[48] E. N. Ganesh, "Development of SMART CITY Using IOT and BIG Data," *Int. J. Comput. Tech.*, vol. 4, no. 1, 2017.

[49] M. R. Bashir and A. Q. Gill, "Towards an IoT Big Data Analytics Framework: Smart Buildings Systems," *Proc. 2016 Ieee 18Th Int. Conf. High Perform. Comput. Commun. Ieee 14Th Int. Conf. Smart City; Ieee 2Nd Int. Conf. Data Sci. Syst.*, pp. 1325–1332, 2016.

[50] A. Cuzzocrea, "Big Data Compression Paradigms for Supporting Efficient and Scalable Data-intensive IoT Frameworks," *Proc. Sixth Int. Conf. Emerg. Databases Technol. Appl. Theory*, pp. 67–71, 2016.

[51] D. Spark, H. P. E. E. Platform, and B. D. Analytics, "Spark – A modern data processing framework for cross platform analytics Deploying Spark on HPE Elastic Platform for Big Data."

[52] A. S. Foundation, "Apache Storm," 2015. [Online]. Available: http://storm.apache.org/index.html. [Accessed: 18-Dec-2017].

[53] T. O. Center, "Introducción a Hadoop y su ecosistema," 2013. [Online]. Available: http://www.ticout.com/blog/2013/04/02/introduccion-a-hadoop-y-su-ecosistema/. [Accessed: 03-Dec-2017].

[54] G. Ingersoll, "Introducing apache mahout: Scalable, commercial friendly machine learning for building intelligent applications," *White Paper, IBM Developer Works*, pp. 1–18, 2009.

[55] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks," *ACM SIGOPS Oper. Syst. Rev.*, pp. 59–72, 2007.

[56] C. L. P. C. and C.-Y. Zhang, *Data-intensive applications, challenges, techniques and technologies: A survey on big data*, Inf. Sci. 2014.

[57] J. Kelly, "Apache Drill Brings SQL-Like, Ad Hoc Query Capabilities to Big Data," *Wikibon*, 2013. [Online]. Available: http://wikibon.org/wiki/v/Apache_Drill_Brings_SQL-Like,_Ad_Hoc_Query_Capabilities_to_Big_Data. [Accessed: 12-Dec-2017].

[58] J. M. and F. W. T. Huang, L. Lan, X. Fang, P. An, "Promises and challenges of big data computing in health sciences," *Big Data Res.*, pp. 2–11, 2015.

[59] S. Maitrey and C. K. Jha, "MapReduce: Simplified Data Analysis of Big Data," *Procedia Comput. Sci.*, vol. 57, pp. 563–571, 2015.

[60] H. Team, "JobTracker and TaskTracker," *Hadoop in real world*, 2015. [Online]. Available: http://hadoopinrealworld.com/jobtracker-and-tasktracker/. [Accessed: 14-Dec-2017].

[61] A. S. Foundation., "Apache Impala Overview," *Cloudera*, 2018. [Online]. Available: https://www.cloudera.com/documentation/enterprise/5-9-x/topics/impala_intro.html#impala_cdh. [Accessed: 08-Jan-2018].

[62] Cloudera, "How-to: Use Apache Solr to Query Indexed Data for Analytics," *Cloudera Engineering Blog*, 2015. [Online]. Available: http://blog.cloudera.com/blog/2015/10/how-to-use-apache-solr-to-query-indexed-data-for-analytics/. [Accessed: 08-Jan-2018].

[63] A. S. Foundation, "Spark 0.8.0," *This document gives a short overview of how Spark runs on clusters, to make it easier to understand the components involved.*, 2014. [Online]. Available: https://spark.apache.org/docs/0.8.0/cluster-overview.html. [Accessed: 09-Jan-2018].

[64] T. Galili, "Five ways to handle Big Data in R," *R news and tutorials contributed by 750 R bloggers*, 2017. [Online]. Available: https://www.r-bloggers.com/five-ways-to-handle-big-data-in-r/. [Accessed: 05-Feb-2018].

[65] BeDataDriven B.V., "Renjin," *Read the Docs*, 2018. [Online]. Available: http://www.bedatadriven.com/products/renjin.html. [Accessed: 16-Feb-2018].

[66] I. de ingeniería del Conocimiento, "7 Herramientas Big Data para tu empresa," 2016. [Online]. Available: http://www.iic.uam.es/innovacion/herramientas-big-data-para-empresa/. [Accessed: 26-Jan-2018].

[67] P. Russom, "Big Data Analytics," *TDWI Best Pract. Rep.*, pp. 1–35, 2011.

[68] H. Inc., "HORTONWORKS," 2017. [Online]. Available: https://es.hortonworks.com/about-us/. [Accessed: 21-Feb-2018].

[69] Cloudera, "Introducing Morphlines: The Easy Way to Build and Integrate ETL Apps for Hadoop," *Cloudera Engineering Blog*, 2013. [Online]. Available: http://blog.cloudera.com/blog/2013/07/morphlines-the-easy-way-to-build-and-integrate-etl-apps-for-apache-hadoop/. [Accessed: 16-Mar-2018].

[70] N. Rajagopalan, "Big Data Analytics with Clickstream," *Digital Transformation & Software Engineering Services*, 2016. [Online]. Available: https://www.ness.com/big-data-analytics-with-clickstream/. [Accessed: 26-Mar-2018].

[71] A. S. Foundation., "Get Started with Hue," *Cloudera*, 2018. [Online]. Available: https://www.cloudera.com/documentation/enterprise/5-9-

x/topics/hue.html#hue_guide_home. [Accessed: 19-Feb-2018].

[72] T. Software, “Build your big data platform with Tableau and Cloudera,” *Tableau and Cloudera*, 2018. [Online]. Available: https://www.tableau.com/tableau-and-cloudera. [Accessed: 23-Mar-2018].

[73] Cloudera, “Tableau for IT-powered Analytics,” 2018. [Online]. Available: https://www.cloudera.com/solutions/gallery/tableau-for-it-powered-analytics.html. [Accessed: 23-Mar-2018].

[74] T. S. Elaine Chen, Product Manager, “Uso de R y Tableau,” 2017. [Online]. Available: https://www.tableau.com/es-es/learn/whitepapers/using-r-and-tableau?ref=wc&signin=69fdb9085869958b26ba49abade2b6e4&reg-delay=TRUE. [Accessed: 22-May-2018].

[75] R. T21, “PREVENCIÓN DEL FRAUDE EN CADENA DE SUMINISTRO,” 2018. [Online]. Available: http://t21.com.mx/logistica/2017/05/09/prevencion-fraude-cadena-suministro. [Accessed: 22-May-2018].

[76] R. I. Kabacoff, “Tree-Based Models,” *Quick-R*, 2017. [Online]. Available: https://www.statmethods.net/advstats/cart.html. [Accessed: 16-Apr-2018].

[77] R. Pandya, J. Pandya, K. P. Dholakiya, and I. Amreli, “C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning,” *Int. J. Comput. Appl.*, vol. 117, no. 16, pp. 975–8887, 2015.

[78] M. H. Tim Kam Ho, “Random forest,” 2018. [Online]. Available: https://en.wikipedia.org/wiki/Random_forest. [Accessed: 16-Apr-2018].

[79] Wikipedia, “Curva ROC,” 2009. [Online]. Available: https://es.wikipedia.org/wiki/Curva_ROC. [Accessed: 22-May-2018].

[80] F. Espinosa, “De datos a dinero en retail: análisis de la canasta de compra,” *Forbes*, 2017. [Online]. Available: https://www.forbes.com.mx/brand-voice/de-datos-dinero-en-retail-analisis-de-la-canasta-de-compra/. [Accessed: 20-Apr-2018].

[81] Michy Alice, “How to Perform a Logistic Regression in R,” *DataSciencePlus*, 2015. [Online]. Available: https://datascienceplus.com/perform-logistic-regression-in-r/. [Accessed: 23-Apr-2018].

[82] J. Cardenas, “Odd, odd ratio... wtf?,” *Networkianos*, 2015. [Online]. Available: http://networkianos.com/odd-ratio-que-es-como-se-interpreta/. [Accessed: 02-Jul-2018].

[83] G. de Operaciones, “Cómo utilizar una Regresión Lineal para realizar un Pronóstico de Demanda,” 2018. [Online]. Available: https://www.gestiondeoperaciones.net/. [Accessed: 23-Apr-2018].

[84] Tableau, “Find Clusters in Data,” 2017. [Online]. Available: https://onlinehelp.tableau.com/current/pro/desktop/en-us/clustering.html. [Accessed: 10-May-2018].

[85] S. D. and S. S. D. P. Acharjya, *Computational Intelligence for Big Data Analysis*. 2015.

[86] A. S. Foundation., “Cloudera Documentation Apache Impala Overview,” *Cloudera*, 2018. [Online]. Available: https://www.cloudera.com/documentation/enterprise/5-9-x/topics/impala_intro.html#impala_cdh. [Accessed: 05-Mar-2018].

Esta página fue intencionalmente dejada en blanco

Anexo A

El apéndice A presenta información adicional sobre el archivo de conformación de Apache Flume. en CDH.

Archivo de configuración flume.conf

```
agent1.sources = source1
agent1.sinks = solrSink
agent1.channels = channel1

# Describe/configure source1
agent1.sources.source1.type = exec
agent1.sources.source1.command = tail -F /opt/gen_logs/logs/access.log
#agent1.sources.source1.command = cat /opt/gen_logs/logs/access.log

# Describe solrSink
agent1.sinks.solrSink.type = org.apache.flume.sink.solr.morphline.MorphlineSolrSink
agent1.sinks.solrSink.channel = memoryChannel
agent1.sinks.solrSink.batchSize = 1000
agent1.sinks.solrSink.batchDurationMillis = 1000
agent1.sinks.solrSink.morphlineFile = /opt/examples/flume/conf/morphline.conf
agent1.sinks.solrSink.morphlineId = morphline
agent1.sinks.solrSink.threadCount = 1

# Use a channel which buffers events to a file
# -- The component type name, needs to be FILE.
agent1.channels.channel1.type = FILE

# The maximum size of transaction supported by the channel
agent1.channels.channel1.capacity = 20000
agent1.channels.channel1.transactionCapacity = 1000

# Amount of time (in millis) between checkpoints
agent1.channels.channel1.checkpointInterval 3000

# Max size (in bytes) of a single log file
agent1.channels.channel1.maxFileSize = 2146435071

# Bind the source and sink to the channel
agent1.sources.source1.channels = channel1
agent1.sinks.solrSink.channel = channel1
```

Esta página fue intencionalmente dejada en blanco

Anexo B

El apéndice B, muestra información adicional sobre la conformación de una Morphline para la indexación de los registros ingeridos por Flume en tiempo real.

Archivo de configuración morphline.conf

```
SOLR_LOCATOR : {
  collection : live_logs       # Nombre de la colección solr
  zkHost : "quickstart.cloudera:2181/solr" # ZooKeeper ensemble
}
morphlines: [{
  id: morphline
  importCommands: ["org.kitesdk.**","org.apache.solr.**"]
  commands : [
    {
      readLine {
        charset : UTF-8
      }
    }
    {
      grok {
        dictionaryFiles : [/opt/examples/flume/morphlines/dictionaries/]
        extract: inplace
        expressions : {
          message : """%{IPORHOST:ip} %{USER:user} %{USER:auth}
\[%{HTTPDATE:request_date}\] "(?:%{WORD:method} %{NOTSPACE:request}(?:
HTTP/%{NUMBER:httpversion})?|%{DATA:full_request})" %{NUMBER:status} (?:%{NUMBER:size}|-
) "(%{DATA:unknown_field})" "(%{DATA:user_agent})""""
        }
      }
    }
    {
      convertTimestamp {
        field : request_date
        inputFormats : ["dd/MMM/yyyy:HH:mm:ss Z", "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'", "yyyy-
MM-dd'T'HH:mm:ss", "yyyy-MM-dd"]
        inputTimezone : America/Los_Angeles
        outputFormat : "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
        outputTimezone : UTC
      }
    }
    {
      java {
        imports : "import java.util.*;import java.util.regex.*;"
        code: """
          String department = "";
          String category = "";
          String product = "";
          String action = "";
          String request_key = record.get("request").get(0).toString();
          if(request_key.equals("/home")) {
            action = "view home";
          }
          Pattern pDepartment = Pattern.compile("/department/(.+?)/");
          Matcher mDepartment = pDepartment.matcher(request_key);
          while (mDepartment.find()) {
            department = mDepartment.group(1);
            action = "view department";
          }
          Pattern pCategory = Pattern.compile("/department/(.+?)/category/(.*)");
```

```
          Matcher mCategory = pCategory.matcher(request_key);
          while (mCategory.find()) {
            department = mCategory.group(1);
            category = mCategory.group(2);
            action = "view category products";
          }
          Pattern pProduct = Pattern.compile("/product/(.*)");
          Matcher mProduct = pProduct.matcher(request_key);
          while (mProduct.find()) {
            product = mProduct.group(1);
            action = "view product";
          }
          Pattern pAddToCart = Pattern.compile("/add_to_cart/(.*)");
          Matcher mAddToCart = pAddToCart.matcher(request_key);
          while (mAddToCart.find()) {
            product = mAddToCart.group(1);
            action = "add product to cart";
          }
          if(request_key.equals("/view_cart")) {
            action = "view cart";
          }
          if(request_key.equals("/checkout")) {
            action = "checkout";
          }
          if(request_key.equals("/support")) {
            action = "support";
          }
          if(request_key.equals("/contact_us")) {
            action = "contact us";
          }
          if(!department.equals("")) {
            record.put("department",department);
          }
          if(!category.equals("")) {
            record.put("category",category);
          }
          if(!product.equals("")) {
            record.put("product",product);
          }
          record.put("action",action);
          return child.process(record);
        """
      }
    }
    {
      generateUUID {
        field : ignored_base_id
      }
    }
    {
      generateSolrSequenceKey {
        baseIdField: ignored_base_id
        solrLocator : ${SOLR_LOCATOR}
      }
    }
    {
      sanitizeUnknownSolrFields {
        solrLocator : ${SOLR_LOCATOR}
      }
    }
    {
       loadSolr {
solrLocator : ${SOLR_LOCATOR}
      }
    }
  ]
}]
```

Anexo C

El apéndice C, indica el código fuente Python del Generador de Registros Web de tiempo real.

Código fuente del programa genhttplogs.py

```
#! /usr/bin/env python
from datetime import datetime
import json
import random
import os
import sys
import time
import urllib

class IPGenerator:
    def __init__(self, session_count, session_length):
        self.session_count = session_count
        self.session_length = session_length
        self.sessions = {}

    def get_ip(self):
        self.session_gc()
        self.session_create()
        ip = self.sessions.keys()[random.randrange(len(self.sessions))]
        self.sessions[ip] = self.sessions[ip] + 1
        return ip

    def session_create(self):
        while len(self.sessions) < self.session_count:
            self.sessions[self.random_ip()] = 0

    def session_gc(self):
        for (ip, count) in self.sessions.items():
            if count >= self.session_length:
                del self.sessions[ip]

    def random_ip(self):
        octets = []
        octets.append(str(random.randrange(223) + 1))
        for i in range(3):
            octets.append(str(random.randrange(255)))
        return ".".join(octets)

class LogGenerator:
    PRODUCTS = {}

    REQUESTS = {
        "/departments": 40,
        "/department/*DEPARTMENT*/categories": 20,
        "/department/*DEPARTMENT*/products": 10,
        "/categories/*CATEGORY*/products": 5,
        "/product/*PRODUCT*": 10,
        "/add_to_cart/*PRODUCT*": 5,
        "/login": 5,
        "/logout": 2,
        "/checkout": 3,
        "/support": 1
    }
    EXTENSIONS = {
        'html': 40,
        'php': 30,
        'png': 15,
        'gif': 10,
        'css': 5,
```

```
}
RESPONSE_CODES = {
    200: 92,
    404: 5,
    503: 3,
}
USER_AGENTS = {
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/35.0.1916.153 Safari/537.36": 11,
    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0": 6,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/35.0.1916.153 Safari/537.36": 5,
    "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/35.0.1916.153 Safari/537.36": 4,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.77.4 (KHTML,
    like Gecko) Version/7.0.5 Safari/537.77.4": 4,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/35.0.1916.153 Safari/537.36": 3,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:30.0) Gecko/20100101
    Firefox/30.0": 3,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.76.4 (KHTML,
    like Gecko) Version/7.0.4 Safari/537.76.4": 3,
    "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/35.0.1916.153 Safari/537.36": 2,
    "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/36.0.1985.125 Safari/537.36": 2,
    "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0": 2,
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/36.0.1985.125 Safari/537.36": 2,
    "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko": 2,
    "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/35.0.1916.153 Safari/537.36": 2,
    "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:30.0) Gecko/20100101 Firefox/30.0":
    1,
    "Mozilla/5.0 (Windows NT 6.1; rv:30.0) Gecko/20100101 Firefox/30.0": 1
}
DEPARTMENTS = {
}
CATEGORIES = {
}
def __init__(self, ipgen):
    self.ipgen = ipgen
    self.set_products()
    self.set_departments()
    self.set_categories()

def set_products(self):
    cwd = os.getcwd()
    json_text = open(cwd + '/data/products.json', 'r').read()
    products = json.loads(json_text)
    for p in products:
        self.PRODUCTS[p['product_id']] = int(5000/p['product_price'])

def set_departments(self):
    cwd = os.getcwd()
    json_text = open(cwd + '/data/departments.json', 'r').read()
    depts = json.loads(json_text)
    for p in depts:
        self.DEPARTMENTS[p['department_name']] = 100/len(depts)

def set_categories(self):
    cwd = os.getcwd()
    json_text = open(cwd + '/data/categories.json', 'r').read()
    cats = json.loads(json_text)
    for p in cats:
        self.CATEGORIES[p['category_name']] = 100/len(cats)

def write_qps(self, dest, qps):
    sleep = 1.0 / qps
    while True:
```

```
            self.write(dest, 1)
            time.sleep(sleep)

    def write(self, dest, count):
        for i in range(count):
            ip = self.ipgen.get_ip()
            request = self.pick_weighted_key(self.REQUESTS)
            product = self.pick_weighted_key(self.PRODUCTS)
            dept    = self.pick_weighted_key(self.DEPARTMENTS)
            cat     = self.pick_weighted_key(self.CATEGORIES)
            request =
urllib.quote(request.replace("*PRODUCT*",str(product)).replace("*DEPARTMENT*",dept).repl
ace("*CATEGORY*",cat).lower())
            ext = self.pick_weighted_key(self.EXTENSIONS)
            resp_code = self.pick_weighted_key(self.RESPONSE_CODES)
            resp_size = random.randrange(2 * 1024) + 192;
            ua = self.pick_weighted_key(self.USER_AGENTS)
            date = datetime.now().strftime("%d/%b/%Y:%H:%M:%S -0800")
            dest.write("%(ip)s - - [%(date)s] \"GET %(request)s HTTP/1.1\" %(resp_code)s
%(resp_size)s \"-\" \"%(ua)s\"\n" %
                {'ip': ip, 'date': date, 'request': request, 'resp_code': resp_code,
'resp_size': resp_size, 'ua': ua})
            dest.flush()

    def pick_weighted_key(self, hash):
        total = 0
        for t in hash.values():
            total = total + t
        rand = random.randrange(total)

        running = 0
        for (key, weight) in hash.items():
            if rand >= running and rand < (running + weight):
                return key
            running = running + weight

        return hash.keys()[0]

ipgen = IPGenerator(100, 10)
LogGenerator(ipgen).write_qps(sys.stdout, 1)
```

Esta página fue intencionalmente dejada en blanco

Anexo D

El apéndice D, detalla los pasos a seguir para la conexión de CDH con Tableau y este a su vez con R.

Conexión y configuración de la fuente de datos

Se inicia Tableau y debajo de **Conectar**, seleccione Cloudera **Hadoop**. Para obtener una lista completa de las conexiones de datos, dirigirse a la opción: **A un servidor**. Luego realizar lo siguiente:

- Dirección ip del servidor en donde se aloja el DW a conectarse: 192.168.218.157. Número de puerto, para este caso: 21050;
- Seleccione el tipo de base de datos: Hive Server 2 o Impala;
- El método de autenticación requerido: Nombre de usuario y contraseña. Coloque en ambos campos: **cloudera**;
- Las opciones de transporte dependen del método de autenticación que se elige y pueden incluir lo siguiente: Binario, SASL o HTTP;
- Para la conexión realizada con CDH, se utiliza el transporte **Binario,** como se ilustra en la Figura 84 y finalmente seleccione: **Iniciar sesión.**

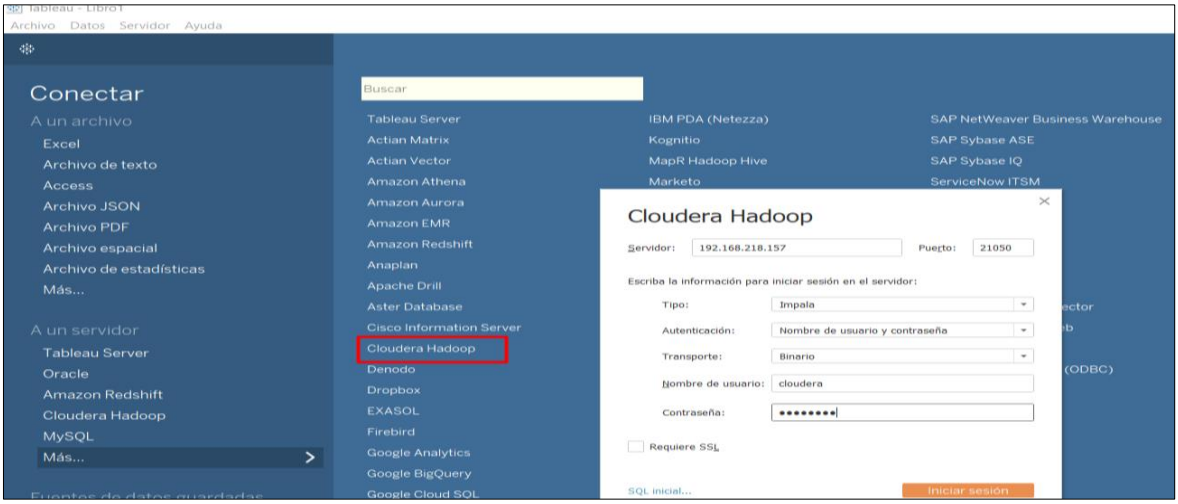


Figura 84. Conexión de Tableau con CDH.

En la página de origen de datos, realizar lo siguiente:

- Seleccionar el nombre de la fuente de datos predeterminado en la parte superior de la página por defecto y luego ingresar el nombre de fuente de datos único para usar en Tableau. En este caso el nombre es: **default;**
- En la lista desplegable **Esquema**, seleccionar el icono de búsqueda o ingresar el nombre del esquema en el cuadro de texto y después el ícono de búsqueda;
- En el cuadro de texto Tabla, seleccionar el ícono de búsqueda e ingresamos el nombre de la tabla;

- Arrastrar la tabla al borde y luego seleccionar la pestaña de la hoja para comenzar su análisis, como se ilustra en la Figura 85.

Figura 85. Configuración del origen de datos en Tableau.

Para la comunicación entre Tableau y R, utiliza un paquete llamado Rserve. Este paquete se instala desde la línea de comandos R, como se indica en la Figura 86, ingresando:

- `install.packages ("Rserve");`
- `library (Rserve);`
- `Rserve()`

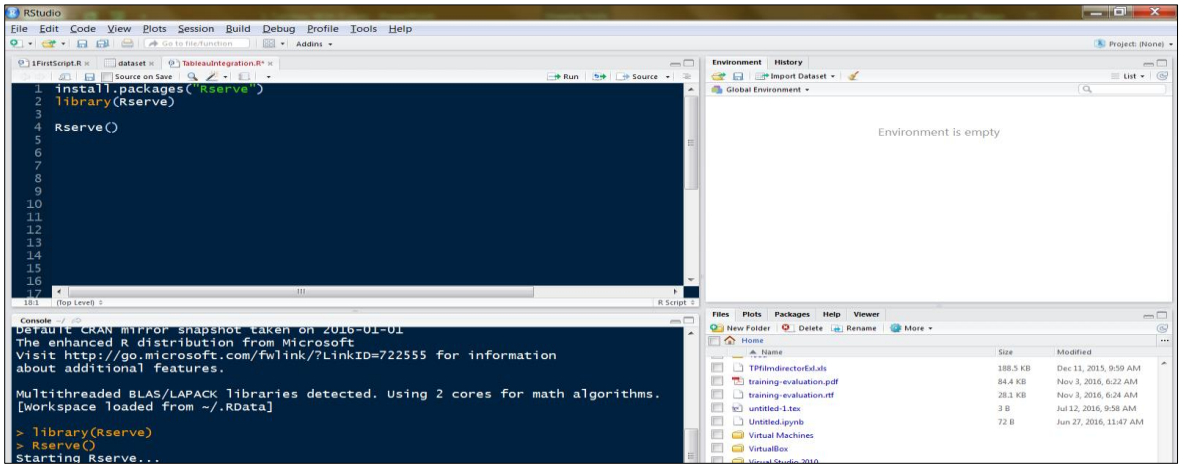


Figura 86. Configuración de R con Tableau.

En el menú “Ayuda de Tableau Desktop”, seleccione “Configuración y rendimiento”, después “Gestionar conexión de R” para abrir el cuadro de diálogo de conexión de **Rserve**. Utilizar Servidor: **localhost** y puerto: **6311**. En este caso no requiere especificar credenciales. Dar clic en el botón “Aceptar”.

Para que Tableau sepa que los cálculos deben ir a R, se debe pasar a través de una de las 4 funciones, las cuales son: SCRIPT_BOOL, SCRIPT_INT, SCRIPT_REAL, SCRIPT_STR. Las funciones R se calculan como cálculos de tabla en Tableau.

Anexo E

El apéndice E, muestra el código fuente en R, para la detección de fraude con *Machine Learning*, en el cual se aplican los modelos: Rpart, C5.0, Random Forest y SVM; realiza una evaluación sobre su rendimiento, tiempo computacional y precisión en la predicción.

Código fuente en R para detección de fraude

```
library(plyr)
library(tidyverse)
library(caret)
library(GGally)
library(stringr)
library(rattle)
library(pROC)
library(ROCR)
set.seed(400)
#Carga del conjunto de datos
fraud_raw <-
read_csv("https://drive.google.com/uc?export=download&confirm=no_antivirus&id=1JRYTW4PkI
p-DTKnhuJfBUOVhBZ_XvStO")
glimpse(fraud_raw)
#Las variables Order_State y Customer_State son técnicamente categóricas
fraud_df <- fraud_raw %>%
  mutate(name_orig_first = str_sub(Order_State,1,6)) %>%
  mutate(name_dest_first = str_sub(Customer_State, 1, 2)) %>%
  select(-Order_State, -Customer_State)
#Prefijos únicos en Customer_State
unique(fraud_df$name_orig_first)
unique(fraud_df$name_dest_first)
#Conversión a factor
fraud_df$name_orig_first <- as.factor(fraud_df$name_orig_first)
fraud_df$name_dest_first <- as.factor(fraud_df$name_dest_first)
table(fraud_df$name_orig_first)
table(fraud_df$name_dest_first)
fraud_df2 <- fraud_df %>%
  select(-name_orig_first) %>%
  select(isFraud, type, hour_month, everything())
glimpse(fraud_df2)
#Las columnas type & isFraud son categóricas y se cambian por factores
fraud_df2$type <- as.factor(fraud_df2$type)
fraud_df2$isFraud <- as.factor(fraud_df2$isFraud)
#Recodificación de factores
fraud_df2$isFraud <- recode_factor(fraud_df2$isFraud, `0` = "No", `1` = "Yes")
summary(fraud_df2)
#Creación de los conjuntos de datos de prueba y entrenamiento y obtención de
transacciones de fraude
fraud_trans <- fraud_df2 %>%
  filter(isFraud == "Yes")
summary(fraud_trans)
#Cuando el tipo es CASH_IN, DEBIT o PAYMENT, hay casos de fraude.
#La cantidad de fraude alcanza un máximo de 1940, se filtran transacciones que superen
esa cantidad.
fraud_df3 <- fraud_df2 %>%
  filter(type %in% c("CASH", "DEBIT","TRANSFER")) %>%
  filter(Sales_per_customer <= 1940) %>%
  select(-name_dest_first)
summary(fraud_df3)
```

```
#Reducción el conjunto de datos principal
not_fraud <- fraud_df3 %>%
  filter(isFraud == "No") %>%
  sample_n(9109)
is_fraud <- fraud_df3 %>%
  filter(isFraud == "Yes")
full_sample <- rbind(not_fraud, is_fraud) %>%
  arrange(hour_month)
#hour_month indica la hora del mes en que se capturaron estos datos, por lo que estos
gráficos se consideran series temporales.
ggplot(full_sample, aes(x = hour_month, col = isFraud)) + geom_histogram(bins = 743)
ggplot(is_fraud, aes(x = hour_month)) +  geom_histogram(bins = 743)
#Con ggpairs se observa si existen correlaciones entre los predictores.
ggpairs(full_sample)
#Verificar la existencia de un patrón de fraude por monto de la transacción
ggplot(full_sample,aes(type, Sales_per_customer,color=isFraud))+geom_point(alpha= 0.01)
+ geom_jitter()
summary(full_sample)
#Pre procesamiento del conjunto de datos completo para modelar
preproc_model <- preProcess(fraud_df3[, -1], method = c("center", "scale", "nzv"))
fraud_preproc <- predict(preproc_model, newdata = fraud_df3[, -1])
#Enlace de resultados a los datos pre procesados
fraud_pp_w_result <- cbind(isFraud = fraud_df3$isFraud, fraud_preproc)
#Resumen de datos pre procesados
summary(fraud_pp_w_result)
#La media de todos los campos numéricos es cero. La desviación estándar es 1
#Selección de columnas numéricas y eliminación de columnas categóricas
fraud_numeric <- fraud_pp_w_result %>%
  select(-isFraud, -type)
high_cor_cols <- findCorrelation(cor(fraud_numeric), cutoff = .75, verbose = TRUE, names
= TRUE, exact = TRUE)
high_cor_removed <- fraud_pp_w_result
#Verificar relaciones lineales entre predictores
fraud_numeric <- high_cor_removed %>%
  select(-isFraud, -type)
comboInfo <- findLinearCombos(fraud_numeric)
comboInfo
#No se identificaron relaciones lineales
#Modelado, copia de datos finales a un marco de datos más general
model_df <-high_cor_removed
#Creación de la misma cantidad de datos de fraude y no fraude para el entrenamiento
is_fraud <- model_df %>%
  filter(isFraud == "Yes")
not_fraud <- model_df %>%
  filter(isFraud == "No")%>%
 sample_n(9109)
# Combinar el conjunto de muestras por 'hour_month'
model_full_sample <- rbind(is_fraud, not_fraud) %>%
  arrange(hour_month)
#División de la muestra para los conjuntos de entrenamiento y prueba
in_train <- createDataPartition(y = model_full_sample$isFraud, p = .75, list = FALSE)
train <- model_full_sample[in_train, ]
test <- model_full_sample[-in_train, ]
gc() #Intervalo para recolección de basura
#Establecer parámetros generales
#Creación de control utilizado para todos los modelos
control <- trainControl(method = "repeatedcv",number = 10, repeats = 3, classProbs =
TRUE, summaryFunction = twoClassSummary)
#Establece un gran conjunto de datos sin fraude
big_no_sample <- model_df %>%
  filter(isFraud == "No") %>%
  sample_n(100000)
#Después de la limpieza y pre procesamiento de datos se procede a aplicar los modelos

#-----------------------------Rpart model-----------------------------------------------
start_time <- Sys.time()
```

```
rpart_model= train(isFraud ~ .,data = train , method = "rpart", tuneLength = 10,
                   metric = "ROC", trControl = control, parms=list(split='information'))
end_time <- Sys.time()
end_time - start_time
#Predicción en el conjunto de entrenamiento
rpart_train_pred <- predict(rpart_model, train)
confusionMatrix(train$isFraud, rpart_train_pred)
#Predicción en el conjunto de prueba
rpart_test_pred <- predict(rpart_model, test)
confusionMatrix(test$isFraud, rpart_test_pred)
#Predicción en un gran conjunto de datos sin fraude
start_time <- Sys.time()
rpart_big_no_pred <- predict(rpart_model, big_no_sample)
end_time <- Sys.time()
end_time - start_time
confusionMatrix(big_no_sample$isFraud, rpart_big_no_pred)
#Trazado de la curva ROC contra los datos de prueba
rpart_probs <- predict(rpart_model, test, type = "prob")
rpart_ROC <- roc(response = test$isFraud,predictor = rpart_probs$Yes,levels =
levels(test$isFraud))
plot(rpart_ROC, col = "blue")
#Área bajo la curva
auc(rpart_ROC)
print(rpart_model)

#-------------------------------------C5.0 model----------------------------------------
#Árboles de decisión y modelos basados en reglas para el reconocimiento de patrones.
grid <- expand.grid( .winnow = c(FALSE), .trials=c(50, 100, 150, 200),  .model="tree" )
start_time <- Sys.time()
c5_model <- train(isFraud ~ .,data = train,method = "C5.0",trControl = control,
                  metric = "ROC", tuneGrid = grid,verbose = FALSE)
end_time <- Sys.time()
end_time - start_time
print(c5_model)
#Predicción en el conjunto de entrenamiento
c5_pred_train <- predict(c5_model, train)
confusionMatrix(train$isFraud, c5_pred_train, positive = "Yes")
#Predicción en el conjunto de prueba
c5_pred_test <- predict(c5_model, test)
confusionMatrix(test$isFraud, c5_pred_test, positive = "Yes")
#Predicción en un gran conjunto de datos sin fraude
start_time <- Sys.time()
c5_pred_big_no <- predict(c5_model, big_no_sample)
end_time <- Sys.time()
end_time - start_time
confusionMatrix(big_no_sample$isFraud, c5_pred_big_no, positive = "Yes")
#Trazado de la curva ROC contra los datos de prueba
c5_probs <- predict(c5_model, test, type = "prob")
c5_ROC <- roc(response = test$isFraud, predictor = c5_probs$Yes, levels =
levels(test$isFraud))
plot(c5_ROC, col = "red")
#Área bajo la curva
auc(c5_ROC)

#------------------------Random Forest model----------------------------------------
grid <- expand.grid(.mtry = 5, .ntree = seq(25, 150, by = 25))
start_time <- Sys.time()
rf_model <- train(isFraud ~ ., data = train, method="rf", metric= "Accuracy",
                  TuneGrid = grid, trControl=control)
end_time <- Sys.time()
end_time - start_time
library(randomForest)
print(rf_model$finalModel)
plot(rf_model$finalModel)
#El error se nivela en alrededor de 100 árboles.
```

```
#Esta gráfica siempre se debe usar para los Bosques Aleatorios para determinar el mejor
punto de corte para los árboles.
varImpPlot(rf_model$finalModel)
#Predicción en el conjunto de entrenamiento
rf_train_pred <- predict(rf_model, train)
confusionMatrix(train$isFraud, rf_train_pred, positive = "Yes")
#Predicción en el conjunto de prueba
rf_test_pred <- predict(rf_model, test)
confusionMatrix(test$isFraud, rf_test_pred, positive = "Yes")
#Predicción en un gran conjunto de datos sin fraude
start_time <- Sys.time()
rf_big_no_pred <- predict(rf_model, big_no_sample)
end_time <- Sys.time()
end_time - start_time
confusionMatrix(big_no_sample$isFraud, rf_big_no_pred, positive = "Yes")
rf_probs <- predict(rf_model, test, type = "prob")
#Trazado de la curva ROC contra los datos de prueba
rf_ROC <- roc(response = test$isFraud,predictor = rf_probs$Yes,levels =
levels(test$isFraud))
plot(rf_ROC, col = "green")
#Área bajo la curva
auc(rf_ROC)

#-------------------------------------SVM model----------------------------------------
start_time <- Sys.time()
svm_model <- train(isFraud ~ ., data = train,
                   method = "svmRadial”, # Kernel Radial
                   tuneLength = 3, metric="ROC", trControl=control)
end_time <- Sys.time()
end_time - start_time
print(svm_model$finalModel)
#Predicción en el conjunto de entrenamiento
svm_train_pred <- predict(svm_model, train)
confusionMatrix(train$isFraud, svm_train_pred, positive = "Yes")
#Predicción en el conjunto de prueba
svm_test_pred <- predict(svm_model, test)
confusionMatrix(test$isFraud, svm_test_pred, positive = "Yes")
#Predicción en un gran conjunto de datos sin fraude
start_time <- Sys.time()
svm_big_no_pred <- predict(svm_model, big_no_sample)
end_time <- Sys.time()
end_time - start_time
confusionMatrix(big_no_sample$isFraud, svm_big_no_pred, positive = "Yes")
#Trazado de la curva ROC
svm_probs <- predict(svm_model, test, type = "prob")
svm_ROC <- roc(response = test$isFraud, predictor = svm_probs$Yes, levels =
levels(test$isFraud))
plot(svm_ROC, col = "black")
#Área bajo la curva
auc(svm_ROC)

#------------------------------Comparación de curvas ROC------------------------------
plot(rpart_ROC, col = "blue")
plot(c5_ROC, col = "red", add = TRUE)
plot(rf_ROC, col = "green", add = TRUE)
plot(svm_ROC, col = "black", add = TRUE)
#Área debajo de las curvas para cada modelo
sort(c(rpart = auc(rpart_ROC), rf = auc(rf_ROC), c5 = auc(c5_ROC), svm = auc(svm_ROC)))
```

Anexo F

El apéndice muestra el código fuente en R, para "*Market Basket Analysis*", utilizando Reglas de Asociación.

Código fuente en R para *Market Basket Analysis*

```
library(tidyverse)
library(arules)
library(arulesViz)
library(plyr)

#Carga del conjunto de datos
retail <-
read.csv("https://drive.google.com/uc?export=download&confirm=no_antivirus&id=1PMwNFQ9CD
ytfv63wdYlBbcilb7VE2wPq", header = T)
retail <- retail[complete.cases(retail), ]
retail %>% mutate(product_name = as.factor(product_name))
retail$order_item_quantity <- as.numeric(as.character(retail$order_item_quantity))
glimpse(retail)
retail %>%
  group_by(order_id ,product_name ) %>%
  summarize(n_items = mean(order_item_quantity))
summary(retail)
head(retail)

# Transforma data.frame en transaccional
trx <- retail
# Convierte datos en lista
trx <- split(trx$product_name,trx$order_id)
trx <- as(trx,"transactions")
# data.frame con frecuencia porcentual de cada producto
FreqProd <- data.frame(Producto=names(itemFrequency(trx)),
                       Frecuencia=itemFrequency(trx), row.names=NULL)
FreqProd <- FreqProd[order(FreqProd$Frecuencia, decreasing = T),]
FreqProd

# Grafica los 10 productos más frecuentes
itemFrequencyPlot(trx,topN=10,type="absolute")

#Extracción de reglas de asociación utilizando el algoritmo Apriori implementado en
#Arules
rules <- apriori(trx, parameter=list(support=0.001, confidence = 0.35))

#Ordena las reglas según la confianza
rules <-sort(rules, by="confidence", decreasing=TRUE)

# Cantidad de reglas creadas
print(rules)

# Imprime todas las reglas
inspect(rules)

# Imprime las 3 reglas de mayor confianza
inspect(head(rules,3))

# Gráfico de dispersión de todas las reglas
plot(rules)
```

```
# Gráfico de grafos de las 14 reglas con mayor confianza
plot(head(rules,14), method="graph", control=list(type="items"))
plot(rules, method="graph")
# Gráfico de matriz de 14 reglas de mayor confianza
plot(head(rules,14), method="grouped")
plot(rules, method="paracoord")
head(quality(rules))

#Se personaliza plot cambiando lift y la confianza
plot(rules, measure=c("support", "lift"), shading="confidence")

#Diagrama de dispersión Two-key plot,
#Support y la confianza son usadas para eje X-Y, el color de los puntos se usa para
#indicar "orden", es decir el número de elementos contenidos en la regla.

plot(rules, shading="order", control=list(main = "Two-key plot"))

#Inspecciona las reglas individuales seleccionándolas
#Inspecciona conjuntos de reglas seleccionando una región rectangular de la gráfica

sel <- plot(rules, measure=c("support", "lift"), shading="confidence", interactive=TRUE)
subrules <- rules[quality(rules)$confidence > 0.35]
subrules
plot(subrules, method="matrix", measure="lift", control=list(reorder=TRUE))
#Una representación alternativa 3D
plot(subrules, method="matrix3D", measure="lift", control=list(reorder=TRUE))
```

Anexo G

El apéndice muestra el código fuente en R, del modelo de “regresión logística binomial" para clasificar y predecir si un envío es tardío (1) o no (0).

Código fuente R para predicción de envíos tardíos

```
library(pscl)
library(ROCR)
library(plyr)
library(tidyverse)
library(caret)

#Carga del conjunto de datos
data.raw  <-
read.csv("https://drive.google.com/uc?export=download&confirm=no_antivirus&id=1AC50nP4-
1MqAjJsXSHTBXGhMx6Fypbj5",header=T,na.strings=c(""))

#verifica los valores perdidos y observa cuántos valores únicos hay para cada variable
sapply(data.raw,function(x) sum(is.na(x)))
sapply(data.raw, function(x) length(unique(x)))

#Selección de columnas relevantes con subset
dataset <- subset(data.raw,select=c(2,3,4,5,6,7,8))
dataset$Late_delivery_risk<-recode_factor(dataset$Late_delivery_risk,’0’="No",’1’="Yes")
not_late <- dataset %>%
  filter(Late_delivery_risk == "No")
is_late <- dataset %>%
  filter(Late_delivery_risk == "Yes")
full_sample <- rbind(not_late, is_late)
ggplot(full_sample, aes(Shipping.Mode, Order.Item.Product.Price, color =
Late_delivery_risk)) +
  geom_point(alpha = 0.01) +
  geom_jitter()

#Codifica las variables categóricas como factores
is.factor(dataset$Customer.Segment)
is.factor(dataset$Shipping.Mode)
is.factor(dataset$Order.Status)

#Montaje del modelo
train <- dataset[1:144420,]
test <- dataset[144421:180519,]
model <- glm(Late_delivery_risk ~.,family=binomial(link='logit'),data=train)
summary(model)

#Se elimina las variables menos significativas del modelo
data <- subset(data.raw,select=c(2,3,8))

#Codifica las variables categóricas como factores
is.factor(data$Shipping.Mode)

#contrasts muestra como R trata las las variables categóricas e interpreta en un modelo.
contrasts(data$Shipping.Mode)

#Montaje del modelo
train <- data[1:144420,]
test <- data[144421:180519,]
model <- glm(Late_delivery_risk ~.,family=binomial(link='logit'),data=train)
summary(model)
confint(model , level=0.95)
```

```
anova(model, test="Chisq")
pR2(model)
exp(coefficients(model))

#Evaluación de la habilidad predictiva del modelo
newdata <- subset(test,select=c(1,2,3))
fitted.results <- predict(model,newdata=subset(test,select=c(1,2,3)),type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != test$Late_delivery_risk)
print(paste('Accuracy',1-misClasificError))

#Trazado de la curva ROC y cálculo del AUC (área debajo de la curva) que son medidas de
#rendimiento para el clasificador binario
p <- predict(model, newdata=subset(test,select=c(1,2,3)), type="response")
pr <- prediction(p, test$Late_delivery_risk)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf, col = "red")
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

Anexo H

El apéndice muestra el código fuente en R, del modelo de “Regresión Lineal Múltiple" utilizado para el pronóstico de la Demanda haciendo uso de información histórica de ventas mensuales de productos.

Código fuente R para pronóstico de la Demanda

```
library(ggplot2)
library(gridExtra)
library(psych)
library(tidyverse)

#Carga del conjunto de datos
datos <-
read.csv("https://drive.google.com/uc?export=download&confirm=no_antivirus&id=1aN3gK5mTs
9wBK7wUtFRe43fUCGp_8ksa", header = T)

#Creación de conjuntos de entrenamiento y prueba a partir de los datos originales.
data <- subset(datos,select=c(1,2,3))
summary(data)
glimpse(data)

#Matriz de Correlación
round(cor(x = data, method = "pearson"), 3)
multi.hist(x = data, dcol = c("blue", "red"), dlty = c("dotted", "solid"),main = "")
set.seed(100)
trainingData<- data[1:27,]
testData <- data[27:37,]

#Desarrollo del modelo con datos de entrenamiento y usado para predecir la distancia en
#los datos de prueba
lmMod <- reg<- lm(Sales ~., data=trainingData)
summary(lmMod)
lmMod

#Intervalo de confianza para cada uno de los coeficientes parciales de correlación
confint(lmMod , level=0.95)
#Validacion de condiciones para la Regresion Lineal Multiple
plot1 <- ggplot(data = trainingData, aes(Month, lmMod$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
plot2 <- ggplot(data = trainingData, aes(Quantity, lmMod$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
grid.arrange(plot1, plot2)

#Distribución Normal de los residuos
qqnorm(lmMod$residuals)
qqline(lmMod$residuals)
shapiro.test(lmMod$residuals)

#Revisión de las medidas de diagnóstico.
#Cálculo del criterio de información Akaike
AIC (lmMod)

#Cálculo de la precisión de predicción y tasas de error
SalesPred <- predict(lmMod, testData)  # predeciccion de distancia
actuals_preds <- data.frame(cbind(actuals=testData$Sales, predicteds=SalesPred))
actuals_preds
```

```
correlation_accuracy <- cor(actuals_preds)
accuracy <-correlation_accuracy
accuracy

# Calculo de la precisión Min Max y MAPE
#MinMaxAccuracy = Average(Min(Actuals,Predicteds) / Max(Actuals,Predicteds))
#MeanAbsolutePercentageError(MAPE) = Average(abs(Predicteds-Actuals) / Actuals))

min_max_accuracy <- mean (apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
# min_max accuracy
min_max_accuracy
mape <- mean(abs((actuals_preds$predicteds -
actuals_preds$actuals))/actuals_preds$actuals)  # Desviacion porcentual absoluta media
mape
```

Glosario

Avro	Formato binario, compacto y rápido de serialización persistente en HDFS que intercambia información entre los nodos del clúster a través de la red. Se basa en esquemas definidos mediante. Los archivos de datos de las tablas en Impala utilizan este formato [61].
BZip	*Codec* de compresión para algunos tipos de archivos, al costo de cierta velocidad al comprimir y descomprimir. Compatible con archivos de texto en Impala 2.0 y versiones posteriores. No compatible con HBase [86].
Confidence	Métrica utilizada en Minería de Asociación que indica el porcentaje de casos donde se da el consecuente respecto a los que cumple el antecedente.
C5.0	Modelo que construye árboles de decisión desde un grupo de datos de entrenamiento y modelos basados en reglas para el reconocimiento de patrones. Los datos de entrenamiento son aumentados con un vector C=c1,c1 …, donde c1 ,c2 representan la clase a la que pertenece cada muestra [77].
Data Warehouse	Un Data Warehouse es un almacén electrónico donde generalmente una empresa u organización mantiene una gran cantidad de información , sus datos son almacenados de forma segura, fiable, fácil de recuperar y fácil de administrar.
Deflate	Codec de comprensión no compatible con archivos de texto [61].
EDW	El depósito de datos empresariales (EDW), es un sistema utilizado para informes y análisis de datos , y se considera un componente central de la inteligencia empresarial.
ETL	Extraer, transformar y cargar (ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos , limpiarlos, y cargarlos en otra base de datos, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.
GZIP	*Codec* de comprensión que utiliza más recursos de CPU que Snappy o LZO, utilizado para datos fríos, a los que se accede con poca frecuencia [61].
Hue	Editor de consultas interactivas basado en la web en la pila de Hadoop que le permite visualizar y compartir datos [71].
Impalad	Daemon que se ejecuta en cada nodo del clúster de Hadoop [61].
Lift	Metrica que mide la proporción de veces en las que los valores del antecedente aparecen juntos frente a los que se daría el consecuente si no estuviesen relacionados.

LZO	*Codec* de compresión solo para archivos de texto y consulta de tablas de texto utilizado por Impala; no permite la creación ni la inserción de datos en tablas [61].
Morphline	Archivo de configuración rico que facilita la definición de una cadena de transformación que consume cualquier tipo de datos de cualquier tipo de fuente, procesa los datos y carga los resultados en un componente de Hadoop [69].
Parquet	Formato de archivo binario orientado a columnas, altamente eficiente para tipos de consultas a gran escala, diseñado para optimizar el almacenamiento y la recuperación de datos de aplicaciones analíticas en Hadoop [61].
Random Forest	Método de aprendizaje conjunto para clasificación , regresión y otras tareas, que operan construyendo una multitud de árboles de decisión en el tiempo de entrenamiento y generando el modo de las clases (clasificación) o predicción media (regresión) de los árboles individuales [78].
RCFile	Formato de archivo que maneja una estructura de ubicación de datos que determina cómo almacenar tablas relacionales en clústeres de computadora. Está diseñado para sistemas que usan el *framework* MapReduce [61].
ROC	Representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. También representa la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) [79].
RPART	Es un modelo de *Machine Learning* de partición recursiva que permite explorar la estructura de un conjunto de datos, visualizar reglas de decisión para predecir un resultado categórico (árbol de clasificación) o continuo (árbol de regresión) ,como los describen Brieman, Freidman, Olshen y Stone)[76].
SequenceFile	Formato de archivo plano que consta de pares clave / valor, binarios. Se usa ampliamente en MapReduce como formatos de entrada / salida. Internamente, los resultados temporales de los mapas se almacenan utilizando *SequenceFile*. Proporciona clases *Writer, Reader* y *Sorter* para escribir, leer y ordenar, respectivamente [61].
Snappy	*Codec* de compresión, recomendado por su equilibrio efectivo entre la relación de compresión y la velocidad de descompresión. Compatible con archivos de texto en Impala 2.0 y versiones posteriores [61].
Support	Es una métrica utilizada en Minería de Asociación que indica el número de casos en el que ocurre el antecedente y el consecuente.
Text	Formato de archivo utilizado para el intercambio con otras aplicaciones o scripts que producen o leen archivos de texto delimitados, como CSV o TSV con comas o pestañas para delimitadores [61].