UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Análisis y Visualización de un Big Data con RHadoop para la toma de decisiones.

TRABAJO DE TITULACIÓN

AUTOR: Merino Jiménez, Santiago Patricio

DIRECTOR: Tenesaca Luna, Gladys Alicia, Mgtr.

LOJA-ECUADOR

2018

Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es

2018

APROBACIÓN DE LA DIRECTORA DEL TRABAJO DE TITULACIÓN

Magister.

Gladys Alicia Tenesaca Luna.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Análisis y Visualización de un Big Data con RHadoop para la toma de decisiones, realizado por Santiago Patricio Merino Jiménez ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, Febrero del 2018.

f) ……

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

"Yo, Santiago Patricio Merino Jiménez, declaro ser autor del presente trabajo de titulación: Análisis y Visualización de un Big Data con RHadoop para la toma de decisiones, de la Titulación de Sistemas Informáticos y Computación, siendo la Ing. Gladys Alicia Tenesaca Luna directora del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad"

f. ……
Autor: Santiago Patricio Merino Jiménez
Cédula: 1105134504

DEDICATORIA

Dedico este trabajo a todas las personas que me acompañaron y ayudaron durante el transcurso de mis estudios universitarios. Principalmente a Dios, al Divino Niño, la Virgencita del Cisne y San Judas Tadeo que gracias a sus bendiciones nunca me dejaron solo y me permitieron seguir en este mundo luego del accidente que sufrí, a mis padres, hermanos, cuñado y sobrinos por su apoyo incondicional, sus consejos y palabras de aliento que me supieron apoyar para seguir superándome cada día y que gracias a ellos no hubiese podido terminar con éxito mis estudios.

A mi abuelita Isabel Jiménez, mi abuelito Julio Merino, mi tía Blanca Jiménez, tío José Masache que ya no se encuentran en este mundo y que gracias a sus consejos, cariño y bendiciones que me supieron brindar cada día que compartí con ellos, los llevo siempre en mi corazón.

A mis abuelitos, tíos y primos que igualmente con sus consejos sabios me enseñaron a no despreocuparme y seguir esforzándome para terminar con éxito mis estudios.

A todos mis amigos que igualmente son como mis hermanos por ayudarme de una u otra manera en cumplir con la finalización de este trabajo y darme palabras de aliento y no dejarme rendir para alcanzar los éxitos.

AGRADECIMIENTO

Agradezco primeramente a Dios, al Divino Niño, la Virgencita del Cisne y San Judas Tadeo por el regalo de la vida y gracias a sus bendiciones me permiten seguir en este mundo para corregir mi vida y seguir compartiendo con mi familia y amigos luego del accidente que sufrí.

A mis padres, Hernando e Hilda, gracias por darme la vida, enseñarme a trabajar, apoyarme siempre en mis estudios, por las bendiciones y los consejos diarios para seguir siendo un hombre de bien y aprovechar todas las oportunidades que se me presenten, a mis hermanos que se encuentran en el exterior, Marco y Rodrigo por su apoyo incondicional, las enseñanzas y consejos como hermanos mayores me supieron brindar, a mi hermana mayor Yadira y mi cuñado Freddy que igualmente me ayudaron con sus consejos y sus compañía cuando tuve la oportunidad de vivir con ellos y gracias por la oportunidad de brindarme de ser tío de mis preciosos sobrinos Martina y Matías que son la adoración de la familia, a mis hermanos menores Diego y Yuliana por su compañía y apoyo diario, espero que este triunfo logrado los llene de satisfacción los quiero con toda mi vida y son mi apoyo incondicional.

A la Universidad Técnica Particular de Loja, que me abrió sus puertas para formarme profesionalmente, igualmente a cada uno de mis docentes que durante toda la carrera universitaria me supieron brindar y compartir sus conocimientos.

A la Ing. Gladys Tenesaca, mi directora de tesis, gracias por su tiempo y enseñanza en la dirección del presente trabajo, sobre todo gracias por su amistad brindada, sus consejos y ánimos para culminar con éxito el mismo. De igual manera gracias a los ingenieros Ramiro Ramírez y Jorge López, por su asesoramiento y revisión.

Finalmente agradezco a mis compañeros y amigos del alma como lo son: Darwin, Juan Carlos, Tito, Jorge, Cristhian, Vanesa, Lizbeth, Alcides, Luis, Gerardo, José y muchos más que siempre estuvieron con sus consejos, enseñanzas y ayuda para salir adelante y culminar con éxito este trabajo.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo de investigación tiene su punto de partida en la utilización de RHadoop como herramienta de análisis, procesamiento y obtención de resultados gráficos de un Big Data.

El presente caso de estudio nace de la necesidad de los gerentes departamentales o institucionales en conocer los resultados hábiles y operativos del estudio de un volumen de datos generado por un sistema, este estudio se lo puede realizar mediante diferentes herramientas, en este trabajo de titulación se lo realiza mediante R y Hadoop, que juntamente integrados se convierten en RHadoop. Hadoop aporta con el procesamiento del Big Data y R con la realización del análisis y generación de visualizaciones, en el desarrollo de este documento se detalla cómo se trabaja con cada una de las herramientas y tiene la finalidad en crear un prototipo que permite la generación de los resultados de Rhadoop, los resultados obtenidos y presentados serán de mucha ayuda para los diferentes gerentes departamentales o institucionales que se encargan de la toma de decisiones.

PALABRAS CLAVES: Big Data, Hadoop, RHadoop.

ABSTRACT

The present research work has its starting point in the use of RHadoop as a tool for analyzing, processing and obtaining graphic results of a Big Data.

This case study is born from the need of departmental or institutional managers to know the skillful and operative results of the study of a volume of data generated by a system, this study can be done through different tools, in this work of titling it is done with R and Hadoop, which together become RHadoop. Hadoop contributes with the processing of Big Data and R with the realization of the analysis and generation of visualizations, in the development of this document it is detailed how it works with each one of the tools and has the purpose in creating a prototype that allows the generation of The results of Rhadoop, the results obtained and presented will be very helpful for the different departmental or institutional managers who are responsible for decision making.

KEYWORDS: Big Data, Hadoop, RHadoop.

INTRODUCCIÓN

Debido a la generación de los datos diariamente a través de la utilización de un sinnúmero de tecnologías, dispositivos móviles, herramientas, sistemas de información, redes sociales, etc., surge la necesidad de que esta información sea almacenada, analizada y procesada para generar valor ya sea a una persona, institución u organización. El análisis se puede realizar mediante la utilización de cualquier herramienta tradicional que permita gestionar este volumen de datos, pero surge el problema que el tamaño del volumen de datos es mucho mayor a la que es posible tratar, almacenar y procesar por las herramientas tradicionales, cuando el volumen de datos no logra ser gestionado por estas herramientas esa información se logra llamar Big Data y necesita de herramientas más avanzadas que tengan la capacidad de proveer dicha gestión.

En base a lo antes mencionado una de las herramientas más conocidas y de mayor uso hoy en día para el procesamiento de un gran volumen de datos es Hadoop, herramienta la cual ofrece en su ecosistema otro tipo de herramientas de libre acceso que se encuentran en constante actualización para proveer la máxima eficiencia en su trabajo. Una herramienta que se logra integrar eficientemente a Hadoop proveyendo sus grandes ventajas de exploración, análisis y visualización de un volumen de datos es R, que integrado exitosamente a Hadoop se convierte en Rhadoop. Esta es la herramienta utilizada en el desarrollo de este trabajo ofreciendo grandes ventajas al realizar el multiprocesamiento del volumen de datos y ofreciendo resultados precisos, que son de mucha ayuda para los gerentes institucionales o departamentales los cuales apoyándose en los resultados obtenidos son los encargados de realizar la correcta toma de decisiones y creación de nuevas estrategias de negocio. Lo más importante del estudio y análisis de un volumen de datos es que proporciona a las instituciones u organizaciones, información que desconocían en la proporción del volumen de datos que poseen, en otras palabras, proporciona un punto de referencia a la identificación de problemas que pueden ser solucionados y convertirlos en estrategias de negocio.

En el desarrollo de este trabajo se describe información relevante al tema de Big Data, Big Data Analytics y como el estudio de Big Data aporta información de suma importancia a los diferentes gerentes de una institución u organización para la toma de decisiones. En el documento también se describe la problemática que lleva a la realización de este trabajo y a las diferentes metodologías que se pueden aplicar para el estudio, procesamiento y obtención de resultados del análisis de un volumen de datos. Finalmente se describe la metodología seleccionada para el desarrollo de la

parte práctica del presente trabajo de titulación, en el que se detalla cada una de las fases que contiene la metodología, posteriormente se realiza cada una de las tareas de análisis y procesamiento del volumen de datos en RHadoop con el cual se genera cada una de las diferentes visualizaciones. Una vez validado el análisis y los resultados de la data se procede a desarrollar un prototipo web en el que se ejecuta un script por cada uno de los resultados que el usuario desea obtener los cuales están diseñadas en Rhadoop y presenta cada una de las visualizaciones obtenidas del análisis realizado, estas visualizaciones servirán de apoyo a las diferentes autoridades que analizarán los resultados obtenidos para la creación de nuevas estrategias de negocio y la toma de decisiones.

De esta manera se ha logrado cumplir con cada uno de los objetivos propuestos del presente trabajo de titulación, iniciando en documentar todo lo relacionado con Big Data, Hadoop RHadoop hasta como realizar el análisis y procesamiento de un volumen de datos mediante RHadoop y el desarrollo de un prototipo web que presente los resultados de forma gráfica de los datos y como estos apoyan a los gerentes a la creación de nuevas estrategias de negocio y realizar la correcta toma de decisiones.

1.1. Objetivos.

Objetivo General.

Analizar y Visualizar un Big Data con RHadoop para la toma de decisiones.

Objetivos Específicos.

- Investigar y conceptualizar Big Data, Hadoop y toma de decisiones.
- Investigar funcionamiento de la herramienta RHadoop.
- Desarrollar visualizaciones de la Big Data con RHadoop para la toma de decisiones.
- Desarrollar prototipo de pruebas de la BigData para la toma de decisiones.

CAPITULO II MARCO TEÓRICO

Figura 1 - Big Data
Fuente: Next – International Business School
Elaboración: Next – International Business School

2.1. Big Data.

Los autores Wu, Zhu, Wu, & Ding, (2013), menciona que Big Data se refiere a grandes volúmenes, conjuntos de datos complejos y en crecimiento con múltiples fuentes autónomas, además que, con el rápido desarrollo de redes, almacenamiento y la capacidad de recopilación de datos, Big Data se está expandiendo rápidamente en todos los ámbitos de la ciencia y la ingeniería, incluidas las ciencias físicas, biológicas y biomédicas. La figura 1, representa el conjunto de palabras que abarca este gran concepto de Big Data, varios autores tienen una definición diferente acerca de este término, a continuación, se presentan las definiciones más relevantes e importantes.

Otros autores como Gill, Nguyen, & Koren, (2009), definen a “Big Data como el conjunto de datos cuyo tamaño va más allá de la capacidad de captura, almacenado, gestión y análisis de las herramientas de base de datos”.

El autor Brynjolfsson, (2012), menciona que la reciente explosión de los datos digitales por medio de la información almacenada es muy importante, porque permite a los gerentes de empresas conocer y medir información relevante a sus negocios y traducir directamente ese conocimiento en la toma de decisiones y el rendimiento mejorados.

Un estudio realizado por, Schroeck, Michael; Shockley, Rebecca; Smart, (2012), consistió en dar a una serie de encuestados opciones sobre como definirían a Big Data, las cuales son: un mayor ámbito de información, nuevos tipos de datos y análisis, información en tiempo real, entrada de datos procedentes de nuevas tecnologías, formas no tradicionales de soportes, grandes volúmenes de datos, la última palabra de moda y datos de redes sociales, entre las cuales deberían escoger dos características que mejor describiera este concepto. El resultado de esta encuesta se hace visible en la Figura 2 “Definición de Big Data”.

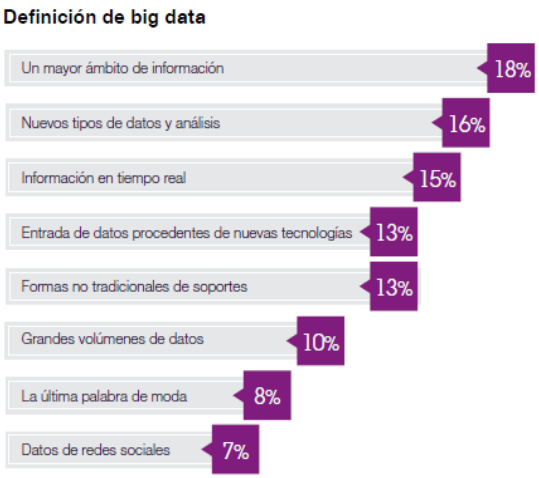


Figura 2 - Definición de Big Data
Fuente: IBM Global Business Services Business Analytics and Optimization
Elaboración: Schroeck, Shockley, Smart.

En conclusión, con lo antes mencionado se determina que Big Data es un volumen de datos que se pueden almacenar, clasificar, analizar, procesar y compartir el almacenamiento masivo de la información mediante la utilización de herramientas especializadas para realizar cada una de las operaciones posibles, dentro de Big Data se determina que se puede encontrar datos estructurados como no estructurados. Con el estudio de los datos se puede encontrar respuestas que permitan tomar decisiones como:

- ✓ Reducción de costes.
- ✓ Reducciones de tiempo.
- ✓ Desarrollo de nuevos productos y ofertas optimizadas.
- ✓ Toma de decisiones inteligentes.

2.1.1. Historia Big Data.

Para describir una breve historia de Big Data se toma como referencia el documento de Suriol, (2014), en el cual se determina que el término “Big Data” fue empleado por primera vez en 1997, cuando un estudio por parte de Michael Cox y David Ellsworth, investigadores de la NASA, utilizaron dicho término para referirse a la gran cantidad de información generada cuando realizaban pruebas de simulación del flujo de aire que se genera alrededor de las naves espaciales en los supercomputadores de la época. En los años 2000 la consultora Gartner define el modelo de las 3V de Big Data en una publicación llamada “Gestión de datos 3D: control de volumen, velocidad y variedad de datos”. Para el año 2004 Google desarrolla MapReduce, al que denominan como un paradigma de procesamiento distribuido, luego de un año Yahoo! desarrolla Hadoop como complemento del motor de búsqueda suyo, el cual se basa en la integración entre MapReduce y HDFS, gracias a esta integración se produce el gran avance del

estudio y explotación de Big Data. En 2005 aparece el término que se conoce como la web 2.0 con lo cual se aumenta el volumen de datos generados por los usuarios que cada día se van integrando con la tecnología y las diferentes redes sociales. IBM determinó que gracias a este avance el 90% de los datos generados a nivel mundial se ha producido en los últimos años.

2.1.2. Las 5V de Big Data.

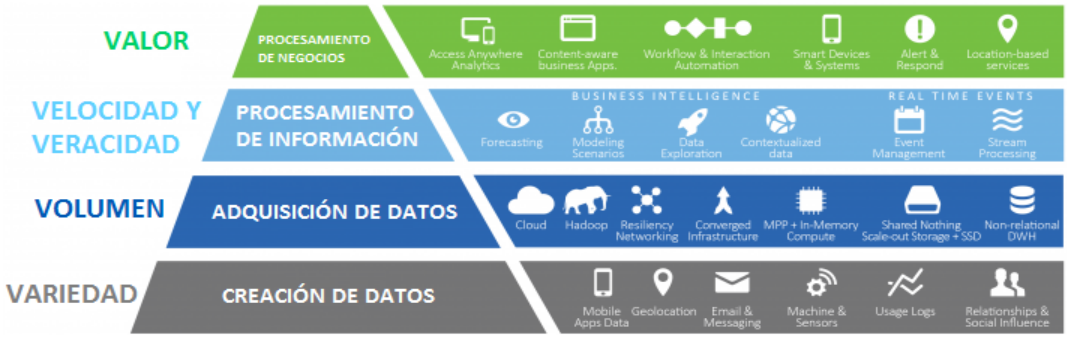


Figura 3 - Las 5V de Big Data
Fuente: EXELACOM
Elaboración: EXELACOM

Los autores Schroeck, Shockley, Smart, (2012), mencionan que Big Data contiene cinco características principales que se conocen como las 5Vs, la figura 3, demuestra el contenido que abarca cada una de las características de las 5Vs de Big Data, y dentro de cada una se demuestra los datos que se pueden encontrar y como se los puede utilizar. Las características se describen a continuación:

2.1.2.1. Valor.

Se refiere a los beneficios que brinda el uso de Big Data (reducción de costes, eficiencia operativa, mejoras de negocio).

2.1.2.2. Velocidad.

Se refiere a la velocidad con la que se producen, procesan y analizan los datos. La velocidad afecta a la latencia, esto es el tiempo de espera entre el momento en el que se generan, capturan y son accesibles los datos. En la actualidad la generación de nuevos datos se produce a una velocidad inimaginable, esto causa que los sistemas utilizados son incapaces de captarlos, almacenarlos y procesarlos.

2.1.2.3. Veracidad.

Se refiere al nivel de fiabilidad de cierto tipo de datos. El esfuerzo que se realiza para conseguir datos de muy alta calidad es un reto muy fundamental de Big Data, pero los métodos de limpieza de datos que existe hoy en día no permiten que los datos sean totalmente confiables.

2.1.2.4. Volumen.

Se refiere a la cantidad de datos que se posee, esta es la característica que identifica a Big Data. El volumen de datos crece a un ritmo sin precedentes.

Lo que constituye la gran dimensión de los datos es en función al tipo de datos que se está acaparando, es decir, si se aplica la recolección de datos en una zona geográfica extensa, la cantidad de datos recolectados va a ser muy grande y puede que estos datos sean de una u otra manera erróneos, según el método de recolección que se aplique.

2.1.2.5. Variedad.

Se refiere a los tipos de datos que podemos encontrar en Big Data, entre los cuales encontramos los datos estructurados, semi-estructurados y los no estructurados.

La mayoría de las organizaciones tienen la necesidad de integrar y analizar los datos obtenidos de diferentes fuentes de información, ya sea de forma tradicional o no tradicional. Con el desarrollo de dispositivos inteligentes, sensores y redes sociales, los datos son generados de diferente forma como: publicaciones en redes sociales, sensores, datos multimedia, etc.

2.1.3. ¿En dónde y para qué se utiliza Big Data?

Big Data es utilizado hoy en día por empresas que manejan gran cantidad de datos, los cuales son generados por el resultado natural del mundo digital actual, además que es importante destacar que los datos masivos no solo sirven como parte cuantitativa, sino que la diferencia y el beneficio se encuentra precisamente en su tratamiento, es decir, en parte cualitativa, ya que nos permite ejecutar una toma de decisiones en línea con los movimientos del mercado.

Big Data permite explotar comercialmente una gran cantidad de datos para crear nuevos servicios de mercado. Hoy en día se acumulan datos en formato digital pero el problema es que estos datos son poco estructurados.

2.1.4. Importancia de Big Data.

Galicia, determina que con toda la información que se encuentra disponible en Internet, las organizaciones empresariales tienen que ser capaces de determinar las interacciones informáticas que realicen sus usuarios, además tienen que poseer la capacidad de recoger, procesar, asimilar y gestionar toda esta información. La recolección de los datos puede generar información muy interesante que permitirá gestionar y crear nuevas oportunidades de negocio.

Los expertos califican que la importancia del análisis de Big Data permite la creación de nuevos servicios, reducción de tiempo y costos empleados en una actividad, incrementar la productividad, una mejor posición en el mercado a diferencia de la competencia, esto no solo implica tratar de poseer o diseñar una enorme base de datos, sino el poder sacar provecho a esos datos generados o almacenados.

2.1.5. Ventajas de Big Data.

Un estudio realizado por Francisco, (2015), determina las ventajas más relevantes que se pueden obtener de la utilización de Big Data, la tabla 1 muestra un resumen:

Tabla 1 - Ventajas de Big Data

Ventaja	Descripción
Análisis de navegación web y hábitos de consumo online.	▪ Mediante el análisis de las redes sociales y los datos de navegación se puede determinar el círculo social con el que interactúa cada cliente, esto permite identificar las necesidades que tiene determinada persona. ▪ La información generada será la más actualizada ya que se genera a cada momento ▪ Se mejora la estrategia mediante el estudio y análisis del consumo de productos y servicios por parte de los clientes lo que a su vez permite la generación nuevas oportunidades de negocio.
Gestión del cambio	▪ Se mejora la estrategia mediante el estudio y análisis del consumo de productos y servicios por parte de los clientes lo que a su vez permite la generación nuevas oportunidades de negocio.
Anticipación a los problemas	▪ Un sistema predictivo de análisis y cruce de datos nos permite prever posibles problemas, como por

	ejemplo una predicción de la caída de las ventas de un negocio.
Mejoras de Procesos	▪ Identificando patrones de fraude, procesos innecesarios y el análisis de la seguridad es posible la detección y simplificación de dichos procesos, lo cual produce un beneficio para la empresa permitiéndole la reducción de operaciones y transacciones sospechosas, esto disminuye los riesgos y costos.

Fuente y Elaboración Propia.

2.1.6. Desventajas de Big Data.

El doctor investigador Moreno, (2014), determina que el mayor reto de la utilización de Big Data es disponer del personal adecuado y con una determinada formación para la ejecución de proyectos de análisis, procesamiento y obtención resultados con datos de gran volumen, para ello se debe poseer la información adecuada, la cual es clave para la obtención de los beneficios esperados.

Entre otras de las desventajas que se pueden mencionar, se describen a continuación:

Tabla 2 - Desventajas Big Data

Desventaja
La principal desventaja es el proceso de adopción de Big Data, es el alto costo de software y hardware que se necesita para hacer posible el correcto manejo de los datos.
El personal de la empresa no se siente conforme con la utilización de Big Data, porque creen que los datos que se obtienen son personales y no deben ser utilizados sin el consentimiento de dicha persona.
La información con la que se puede contar puede estar desactualizada.

Fuente y Elaboración Propia.

2.1.7. Fuentes y Tipos de datos de Big Data.

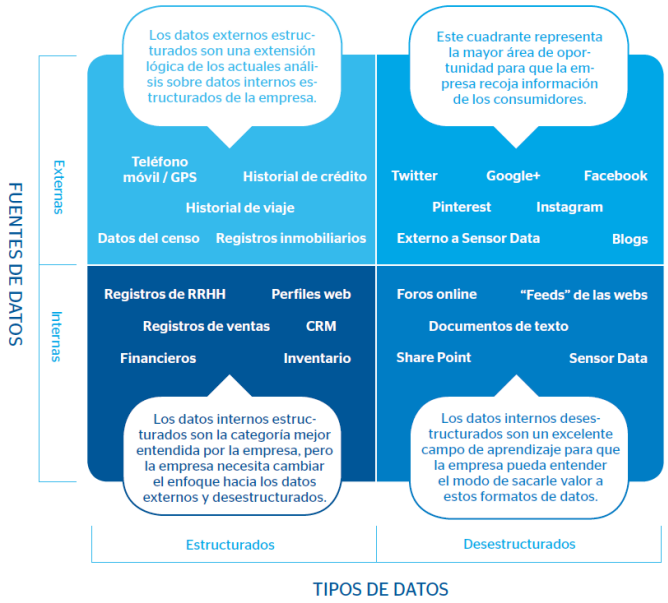


Figura 4 - Fuentes y Tipos de Datos
Fuente: Booz & Company | Benefitting from Big Data
Elaboración: Booz & Company | Benefitting from Big Data

La revista de tecnología BBVA - Innovation Center, (2013) determina que los datos generados para alimentar a Big Data se obtienen de diferentes fuentes ya sean internas o externas, las fuentes internas que generan datos son todos los tipos de registros financieros, ventas, recursos humanos, perfiles web, foros, documentos de texto, etc., y los datos de fuentes internas generadas por los mismos teléfonos móviles, historial de diferente tipo como de créditos, viajes, datos de censos, registros inmobiliarios y las diferentes redes sociales como Twitter, Facebook, Instagram, etc. Los datos con los que nos encontramos en un Big Data pueden ser de diferente tipo y provenir de diferentes fuentes como lo muestra la Figura 4.

Los tipos de datos que se puede encontrar en un Big Data se los divide en:

2.1.7.1. Datos Estructurados.

Son aquellos datos en los que se identifica su longitud y formato, como números, fechas, cadenas de caracteres y se encuentran almacenan en tablas. Un ejemplo de estos son las bases de datos relacionales y hojas de cálculo.

2.1.7.2. Datos no Estructurados.

Son los datos que carecen de un formato específico, por lo que se encuentran en el formato tal y como se recolectaron. No están contenidos en una base de datos o tipo de estructuras de datos. Se generan en mensajes de correo electrónico, documentos de texto, PDFs, software de colaboración y documentos multimedia.

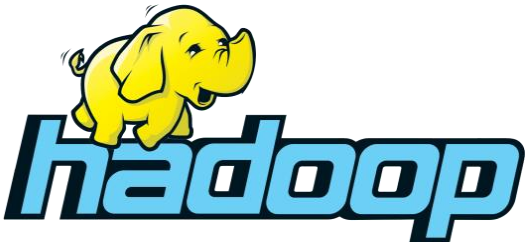


Figura 5 - Hadoop
Fuente: Hadoop and the hype!!
Elaboración: Hadoop and the hype!!

2.2. Hadoop.

Los autores Bagwari & Kumar, (2017), determinan que Hadoop es un framework que sirve para almacenar y procesar gran cantidad de datos, en donde su componente HDFS almacena los datos de forma distribuida preservando su consistencia y disponibilidad mientras que MapReduce es responsable del procesamiento paralelo. Hadoop se adapta mejor al almacenamiento tolerante a fallos y procesamiento por lotes, pero la búsqueda no está optimizada en Hadoop ya que almacena datos en forma de bloques. Carece de un diseño de índice optimizado que conduce a un costoso mecanismo de búsqueda. Para hacer frente a estos diversos enfoques de indexación se han propuesto como una mejora en la arquitectura Hadoop. La figura 5, un elefante amarillo es la representación mundial de Hadoop.

Con lo antes mencionado se puede determinar que:

- Hadoop es un framework de código abierto desarrollado por Apache en el lenguaje de programación Java.
- Hadoop permite almacenar y procesar gran cantidad de datos en un entorno distribuido en clústeres de ordenadores utilizando modelos de programación simple.
- Está diseñado para que un sistema de único servidor se pueda extender a miles de máquinas, esto le permite que sea muy tolerante a los fallos, esto conlleva a brindar la ventaja de que en vez de utilizar hardware costoso.

2.2.1. Historia de Hadoop.

La revista tecnológica SAS realiza una publicación de una breve historia del nacimiento de Hadoop, la cual se resume en la línea de tiempo representada en la figura 6.

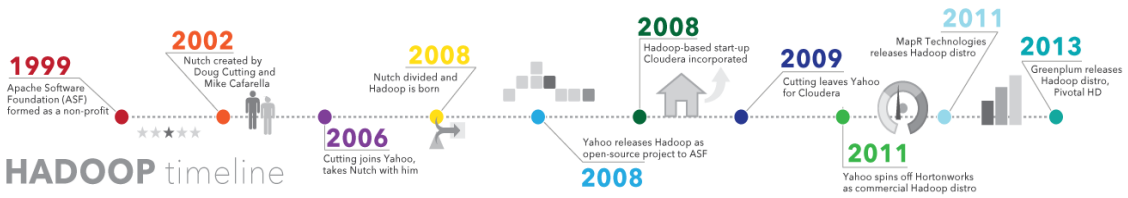


Figura 6 - Historia Hadoop.
Fuente: Revista SAS.
Elaboración: Revista SAS.

Los autores Patel & Singh, (2017), señalan que Hadoop nace cuando Google tiene la necesidad de poseer una solución que le permita continuar procesando datos de manera muy acelerada, por la gran cantidad de datos que maneja su buscador. El problema de Google era poder indexar la web al nivel que exige el mercado y por ello busca una solución que se basa en un sistema de archivos distribuidos.

Esta solución se basa en que un gran número de computadores se encargue de procesar cierta parte de información de forma individual, en donde la gran ventaja del funcionamiento de este sistema es que, la información es dividida y enviada a cada computador del sistema distribuido y cada computador de este sistema maneja y procesa la información recibida de forma independiente y autónoma, pero al final del procesamiento se devuelve el resultado en donde todos actúan en conjunto, como si fueran una sola supercomputadora; esto conllevaría después a lo que denominará Hadoop.

Es en el año 2006, Google decide publicar detalles acerca de su nuevo descubrimiento, en la que comparte su experiencia y conocimiento con los usuarios que deseaban acceder a esta información. Entre todos los que se benefician por el descubrimiento de Google se encuentra la comunidad Open Source. Luego de todo esto, Yahoo! toma el relevo de este descubrimiento impulsando su expansión, que le permitan alcanzar a grandes empresas como Facebook.

2.2.2. Características de Hadoop.

Las características más importantes de Hadoop se describen en la tabla 2, para lo cual se han tomado varios documentos de referencia como lo son el estudio realizado por Sethia, Sheoran, & Saran, (2017) y otro estudio realizado y publicado por Chen, (2017), con los cuales se concluye:

Tabla 3 Características Hadoop

Característica	Beneficio
Escalabilidad y Rendimiento	Procesamiento localmente distribuido de datos para cada nodo del clúster de Hadoop, ya que permite almacenar, gestionar, procesar y analizar datos a escala de petabytes.
Flexibilidad	Se puede almacenar datos en cualquier tipo de formato, ya sea los formatos semi-estructurados o no estructurados, cuando se leen estos datos son analizados y luego se les aplica un esquema.
Bajo costo	Hadoop es de código abierto y se ejecuta en hardware de bajo costo.
Tolerancia a Fallos	Los datos y procesamiento de la aplicación están protegidos contra errores de hardware. Si un nodo se desactiva, los trabajos se redirigen automáticamente a otros nodos para asegurarse de que la computación distribuida no falla. Múltiples copias de todos los datos se almacenan de forma automática.
Poder Computacional	El modelo de computación distribuida que posee Hadoop procesa grandes volúmenes de datos de manera rápida.

Fuente y Elaboración Propia.

2.2.3. Arquitectura de Hadoop.

La figura 7, representa la arquitectura con la que cuenta Hadoop, en la cual podemos reconocer sus cuatro componentes que son MapReduce, HDFS, YARN y las Utilidades Comunes.

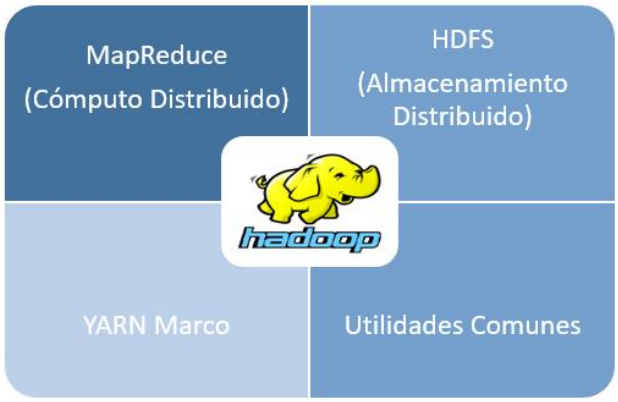


Figura 7 - Arquitectura Hadoop
Fuente: Autor.
Elaboración: Autor.

2.2.3.1. MapReduce.

Para la elaboración de este apartado se ha tomado en cuenta dos estudios realizados por Dean & Ghemawat, (2010), y Chu et al., (2007), con lo cual se describe lo siguiente.

MapReduce se lo puede describir que es un paradigma de programación que proporciona un sistema de procesamiento de datos en paralelo y distribuido, el cual se divide en dos fases que son: Map, la cual se encarga del mapeo y se aplica a cada elemento de la entrada de datos y Reduce, que se encarga de recibir los valores intermedios procesados para agruparlos y producir el resultado final; se dice que MapReduce es el corazón de Hadoop y que fue creado en el año 2004 en Google por Jeffrey Dean y Sanjay Ghemawat, y fue utilizado para calcular el algoritmo de PageRank de Google.

2.2.3.1.1. *Arquitectura MapReduce.*

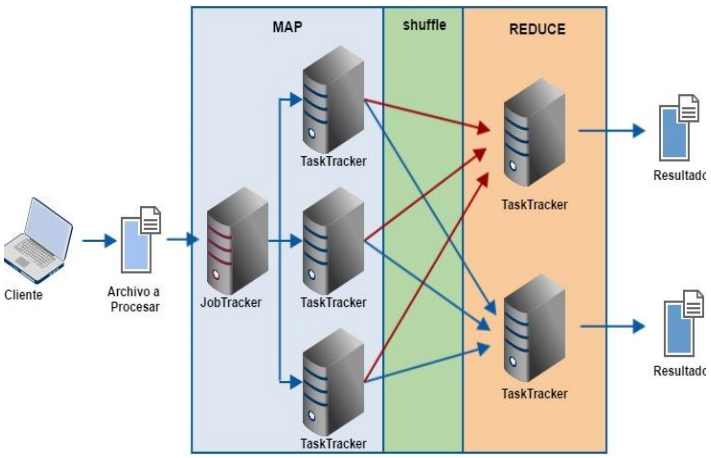


Figura 8 - Arquitectura MapReduce
Fuente: Autor.
Elaboración: Autor.

La figura 8, representa la arquitectura que posee MapReduce, a la cual se la denomina con el nombre de maestro/esclavo, y sus componentes se describen a continuación:

- **Cliente:** es el único componente que puede poner en funcionamiento el proceso enviando un trabajo, ya que es el iniciador.
- **Organizador de intercambios en el sistema distribuido:** es un elemento principal debido a la importancia que tiene al distribuir el trabajo, se encarga de la distribución del trabajo entre las distintas entidades.
- **JobTracker:** es el encargado de coordinar todo el trabajo. Existe un único JobTracker por cada cluster, está encargado de recibir las peticiones de los clientes y organizar el trabajo para los TaskTrackers. Para la elección del TaskTracker, el JobTracker tiene en cuenta el estado de disponibilidad que

tengan los TaskTrackers y si es que se encuentran en el mismo rack. El JobTracker durante todo el tiempo que está realizando el trabajo no pierde contacto con los TaskTracker. Todos los TaskTracker tienen la obligación de enviar un paquete de control cada determinado tiempo para tener informado al JobTracker sobre su estado.

- **TaskTracker:** Están encargados de realizar las tareas en que se ha dividido el trabajo mediante la creación de los diferentes hijos que trabajan en paralelo.

2.2.3.1.2. *Características.*

Entre las principales características de MapReduce podemos listar las siguientes:

- MapReduce permite el procesamiento distribuido de las operaciones de mapeo y reducción, siendo cada operación de mapeo independiente de las otras, las cuales pueden ser realizadas en forma paralela.
- MapReduce puede ser aplicado a grandes conjuntos de datos que son procesados por servidores comunes.
- MapReduce trabaja de manera igual que una base de datos, por motivo que permite almacenar y obtener datos, a diferencia que lo hace de una manera más apropiada para el manejo de grandes cantidades de datos del tipo no estructurado

2.2.3.1.3. *Funcionamiento.*

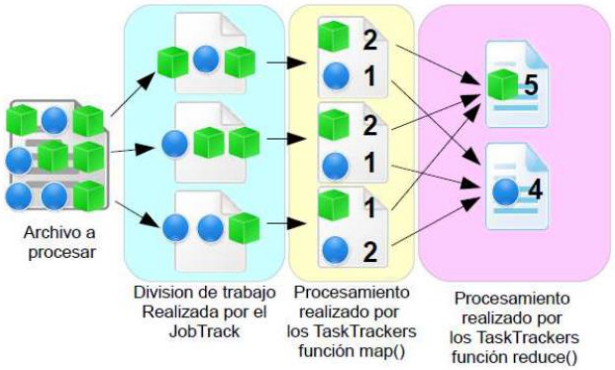


Figura 9 - Funcionamiento de MapReduce
Fuente: SolidQ
Elaboración: SolidQ

El funcionamiento de MapReduce, consiste en poseer un servidor maestro o JobTracker y varios servidores esclavos o TaskTrackers, uno por cada nodo del clúster, como se muestra en la Figura 9, para mayor entendimiento se describen a continuación:

- El JobTracker es el punto de interacción entre los usuarios y MapReduce. Los usuarios envían trabajos MapReduce al JobTracker, el cual los ordena en una

cola de trabajos pendientes y los ejecuta según como fueron llegando, además el JobTracker se encarga de gestionar la asignación de tareas y delega las tareas a los TaskTrackers.

- Los TaskTrackers ejecutan tareas bajo la orden del JobTracker y también manejan el movimiento de datos entre la fase Map y Reduce.

2.2.3.1.4. Problemas.

Una vez ya descrito que es MapReduce, y conocer cuál es su arquitectura es hora de hablar de los problemas que encontramos dentro de este componente.

El principal problema que presenta MapReduce es en su capa de funcionalidad ya que presenta falta de memoria al momento de procesar gran cantidad de datos, también se ha identificado que este problema se asocia al hardware utilizado.

Para solucionar estos problemas de caídas de nodos en el TaskTracker, MapReduce tiene la capacidad de activar otro nodo que esté disponible sin ningún trabajo, para que así se pueda continuar la operación que el anterior nodo no ha podido concluir con satisfacción.

2.2.3.2. HDFS.

Para la elaboración de este apartado se ha tomado en cuenta dos estudios realizados por Borthakur, (2008), Shvachko, (2010), y Kala Karun & Chitharanjan, (2013), con lo cual se puede determinar lo siguiente.

HDFS se define como “un sistema de ficheros distribuido, escalable y portátil, que fue desarrollado en Java y fue diseñado para trabajar con ficheros de gran tamaño”.

Otra definición que se puede dar de HDFS es como un sistema de archivos distribuido, el cual se encarga de proporcionar el acceso de alto rendimiento a los datos a través de grupos de Hadoop, últimamente HDFS se ha convertido en una herramienta muy trascendental para la gestión de los grandes volúmenes de datos y apoyar al análisis de grandes volúmenes de datos de aplicaciones. Su arquitectura se muestra en la Figura 10.

2.2.3.2.1. *Arquitectura HDFS.*

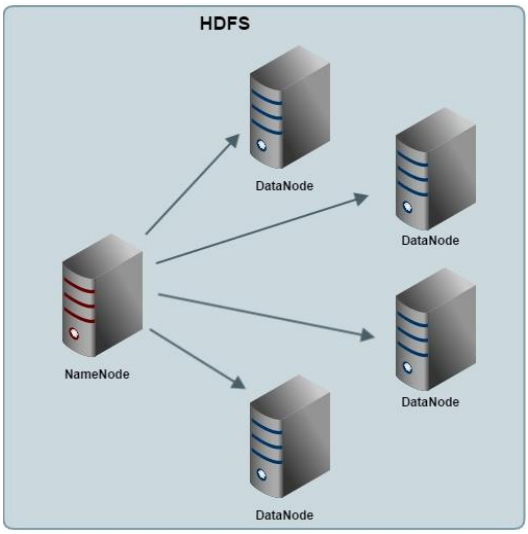


Figura 10 - Arquitectura HDFS
Fuente: Autor.
Elaboración: Autor.

La arquitectura con la que cuenta HDFS, la cual se encuentra distribuida de la siguiente manera:

- Datanode o tambien llamado slave: Este componente es el cual contiene los bloques de información. Tienen capacidad de ser multitarea y es el encargado de los procesos de lectura/escritura. Los DataNodes en ciertos casos son réplicas de otros, cada DataNode tiene una identificacion de almacenamiento unica, lo que le permite ser identificado de un NameNode, es capaz de soportar entre 10 y 4000 DataNodes.
- Namenode o tambien llamado master, es aquel que se encuentra encargado del cierre, apertura y renombrado de directorios y ficheros. Ademas tiene el control y la información sobre la asignación de los bloques en los DataNodes. En caso de pérdida de una réplica será el encargado de replicarla de nuevo en otro DataNode.

2.2.3.2.2. *Características.*

Las características de HDFS son descritas en la siguiente tabla:

Tabla 4 Características HDFS

Característica	Descripción
Escalabilidad	Puede almacenar una cantidad ilimitada de datos en una sola plataforma, según van creciendo los datos se puede añadir más servidores de acuerdo a las necesidades.

Flexibilidad	Capacidad de almacenar datos de cualquier tipo. El ser flexible significa que siempre tendrá acceso a los datos con fidelidad completa para una amplia gama de análisis y casos de uso.
Confiabilidad	Confiabilidad determina que los datos siempre estén disponibles al acceso y sean tolerantes a la pérdida de los mismos, esto significa que los servidores pueden fallar en cualquier momento por cualquier imprevisto, pero que su sistema permanecerá disponible para todas las cargas de trabajo.

Fuente y Elaboración Propia.

2.2.3.2.3. *Funcionamiento.*

El funcionamiento de HDFS es la distribución de información a sus diferentes DataNode, así la información no se encuentra almacenada en un solo NameNode, la gran ventaja de este funcionamiento es que, al almacenar los datos en diferentes DataNode, el sistema es tolerante a la posible pérdida de datos en caso de que se produzca un fallo.

2.2.3.2.4. *Problemas.*

Entre los problemas que se pueden identificar en HDFS uno de los principales es en el momento de acceder a un DataNode, estos fallos se producen porque el nodo buscado ya no se encuentra en el DataNode.

Para tener menos posibilidades de que esto ocurra, se debe procurar facilitar la aplicación cliente en un bloque del que se tiene una constancia sobre su integridad. Para lograr esto los DataNodes deben ser escaneados regularmente y se deben sus checksums y éstos posteriormente ser almacenados en un informe, así se logrará tener constancia sobre cuál fue el último momento de verificación de cada DataNode y por lo tanto saber cuáles han sido revisados últimamente.

2.2.3.3. YARN.

Para la elaboración de este apartado se ha tomado en cuenta dos estudios realizados por Patil & Phatak, (2014), Douglas, Lowe, Malley, & Reed, (2013) y Yao, Wang, Sheng, Lin, & Mi, (2014), con los cuales se puede determinar lo siguiente.

YARN (Yet Another Resource Manager) es conocido como la evolución de MapReduce, también llamado MapReduce 2.0, es un framework de Hadoop, que es distribuido por Apache desde el 2012. Su principal cambio con respecto a MapReduce es la separación de las tareas que antes realizaba el JobTracker, por ser un módulo separado ahora, el Resource Manager es el encargado de supervisar y negociar los

recursos. Básicamente es el encargado de organizar el trabajo y recibir las peticiones de los clientes.

2.2.3.3.1. *Arquitectura YARN***.**

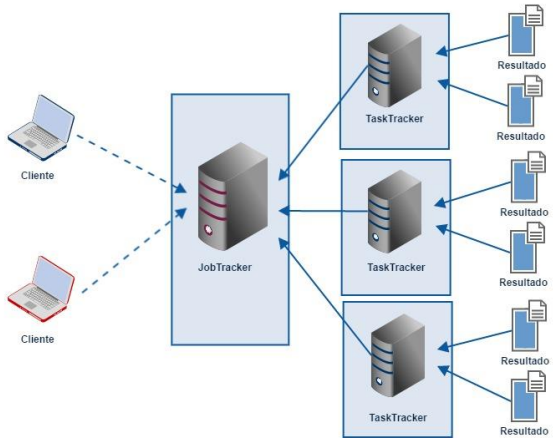


Figura 11 - Arquitectura YARN
Fuente: Autor.
Elaboración: Autor

YARN es utilizado para dividir las responsabilidades de la gestión de los recursos, que son el JobTracker y trabajo de programación/monitorización, en componentes separadas que se llamarán: ResourceManager global y ApplicationMaster, para mayor entendimiento de su funcionamiento se tiene la figura 11, y a continuación se describe cada uno de sus componentes. El ResourceManager es la autoridad que controla los recursos entre todas las aplicaciones del sistema.

El NodeManager es el esclavo por equipo, el cual tiene la responsalbilidad del lanzamiento de los contenedores de las aplicaciones, el seguimiento del uso de recursos del sistema como lo son el CPU, memoria, disco, red, e informar todo a la ResourceManager.

El ApplicationMaster es responsable de negociar los contenedores de recursos adecuados desde el programador, el seguimiento de su estado y el seguimiento de los progresos.

2.2.3.3.2. *Características***.**

Entre las principales características de YARN tenemos:

Ayudar a Hadoop a tener un entorno de gestión de recursos y aplicaciones distribuidas en el cual se pueden implementar múltiples aplicaciones de procesamiento de datos totalmente personalizadas y específicas para realizar una tarea en cuestión.

Permitir al usuario interactuar con todos los datos de múltiples maneras a la vez, por lo que convierte a Hadoop en una auténtica plataforma de datos multiuso, permitiéndole alcanzar su puesto en una arquitectura de datos moderna.

2.2.3.3.3. *Funcionamiento*.

El funcionamiento de YARN se puede describir de tal manera que el ResourceManager y el NodeManager de cada nodo son los encargados de formar el entorno de trabajo, teniendo la responsabilidad el ResourceManager de repartir y gestionar los recursos entre todas las aplicaciones del sistema mientras que el ApplicationMaster es el encargado de la negociación de recursos con los elementos del ResourceManager y los NodeManager para poder ejecutar y controlar las tareas, todo esto se resume en que se encarga de solicitar recursos para poder trabajar.

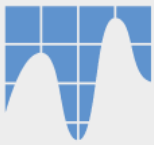
2.2.4. Funcionamiento de Hadoop.

Los autores Bagwari & Kumar, (2017), mencionan que Hadoop funciona de manera en que cada nodo de su estructura contiene solamente un nodo de datos, un clúster de datos conforma el clúster HDFS. El funcionamiento normal es sencillo ya que cada nodo no necesita de un nodo de datos para poder estar en constante funcionamiento, esto se produce porque cada nodo sirve de bloque de datos sobre la red usando un protocolo de bloqueo específico para HDFS. A su vez HDFS tiene la capacidad de almacenar archivos sumamente grandes a través de múltiples máquinas, con lo cual logra alcanzar alta fiabilidad mediante el replicado de los datos a través de múltiples hosts, estos datos se logran almacenar en 3 nodos gracias a el valor por defecto de replicación que es 3; de estos 3 nodos dos se almacenan en el mismo rack y el restante en un rack diferente. Estos nodos pueden estar en constante comunicación para poder actualizar y equilibrar sus datos, realizar copias y conseguir replicación de datos. El HDFS y MapReduce son las partes más importantes, por lo que son los encargados de procesar altas cargas de trabajo con gran éxito.

2.2.5. Importancia de Hadoop.

Para definir la importancia de Hadoop se lo realizará mediante la presentación de la tabla 5, en la cual se detallará las características más importantes que se pueden presentar de Hadoop, según un artículo presentado por IBM (2013).

Tabla 5 Importancia Hadoop

 **Despliegue en Cualquier Lugar** La estructura de los datos los cuales se encuentran sin ningún esquema permite a Hadoop añadir e integrar múltiples datos de diferentes orígenes y de diferentes estructuras.	 **Escalable** Hadoop permite añadir capacidades sin tener la necesidad de cambiar los formatos de los datos, cómo se cargan los datos o cómo se escriben los trabajos o las aplicaciones.
 **Herramientas Avanzadas** Hadoop dispone de una gran variedad de herramientas las cuales permiten presentar gran tipo de visualizaciones, aprendizaje de máquina, análisis del texto y más.	 **Rentable** Todas las herramientas que presenta Haddop presentan gran rentabilidad y así confiar que las visualizaciones, el aprendizaje de máquina, el análisis del texto sean muy útiles.
 **Almacenamiento** Con la arquitectura flexible infinitamente escalable de Hadoop, gracias a que está basado en HDFS, permite a las empresas almacenar y analizar cantidades ilimitadas de datos, todo ello en una única plataforma.	 **Proceso** Hadoop permite la integración rápidamente con los sistemas o aplicaciones existentes para mover datos dentro y fuera de Hadoop a través del procesamiento de carga a granel o streaming.

Fuente: Elaboración Propia

2.2.6. Ventajas de Hadoop.

Un estudio realizado por K Shvachko, (2010), determina los beneficios de utilizar Hadoop, las cuales son:

Tabla 6 Ventajas de Hadoop

Escalabilidad y rendimiento	• Se refiere al procesamiento distribuido de datos para cada nodo en un clúster Hadoop, lo cual le permite almacenar, gestionar, procesar y analizar datos.
Fiabilidad	• Se refiere a que los equipos presenten fallas de sus nodos individuales, entonces Hadoop es fundamentalmente

	tolerante a estos fallos de procesamiento.
Bajo costo	• Al ser software libre Hadoop tiene la ventaja de ejecutarse en cualquier equipo que posea hardware de bajo costo.

Fuente y Elaboración Propia.

2.2.7. Desventajas de Hadoop.

Una vez que se ha descrito lo que es Hadoop, así como su historia, características, arquitectura, su funcionamiento e importancia, es hora de hablar sobre sus desventajas, las cuales son pocas, pero es necesario mencionarlas, ya que estas se presentan por su arquitectura, esto se describe en la tabla 7, para su elaboración se toma en cuenta un artículo de Zhou, (2013).

Tabla 7 Desventajas de Hadoop

HDFS	• Latencia producida por el acceso a los datos, ya que HDFS se encuentra está orientado a procesos batch y operaciones en streaming. esto produce que la latencia de cualquier operación IO no ha sido optimizada y los sistemas de archivos tradicionales suelen ser más rápidos en estos aspectos. • La gran cantidad de ficheros pequeños que posee hace que el límite en el sistema está limitado por la memoria del NameNode, ya que en su memoria RAM es donde se encuentran los metadatos y en donde cada fichero, directorio y bloque ocupa un tamaño de entre 150 y 200 bytes, esto hace que si existen gran cantidad de ficheros pequeños va a ocupar mucho más espacio en la RAM que si se tiene menos cantidad de ficheros de gran tamaño.
MapReduce	• Su depuración es muy complicada ya que al procesar el programa en los nodos donde se encuentran los bloques de datos, no es fácil encontrar los fallos de código. Tampoco es conveniente utilizar funciones de escritura de logs en el código ya que eso podría suponer un gran aumento en la ejecución de procesos MapReduce. • No todos los algoritmos se pueden escribir con el paradigma MapReduce. • Latencia producida por que cualquier trabajo que realiza MapReduce suele tardar alrededor de 10 segundos, por lo que si el volumen de información a tratar es pequeño, es posible que Hadoop no sea la solución más rápida.

Fuente y Elaboración Propia.

2.2.8. RHadoop.

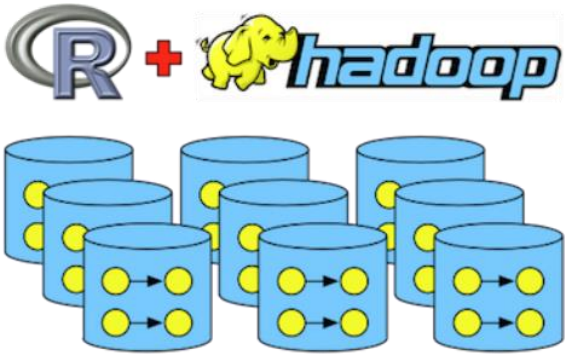


Figura 12 - RHadoop
Fuente: Rose Technologies.
Elaboración: Rose Technologies.

RHadoop es la combinación de dos tecnologías como lo son R y Hadoop, desarrollado por el equipo de Revolution Analytics, estas tecnologías se complementan eficientemente, permitiendo el análisis y la visualización de grandes volúmenes de datos. Los autores Yu-Wei & Chiu, (2015), definen que “RHadoop es una colección de R paquetes que permite a los usuarios procesar y analizar grandes volúmenes de datos con Hadoop”.

Mientras tanto Worms, (2012), define que “RHadoop es un puente entre R, un lenguaje y entorno para explorar estadísticamente conjuntos de datos, y Hadoop, un marco que permite el procesamiento distribuido de grandes conjuntos de datos a través de clústers de computadores.” Además, menciona que RHadoop se construye a partir de 2 paquetes de R que son: rhdfs que se encarga de almacenar los datos al HDFS para posterior trabajarlos con el rmr que proporciona una forma de que los analistas de datos accedan a un procesamiento masivo tolerante a los fallos sin necesidad de dominar la programación distribuida. Estos paquetes son desarrollados primordialmente para sistemas de Cloudera y Hortonworks, los cuales son sistemas desarrollados específicamente para trabajar con Hadoop, los paquetes desarrollados deben tener una amplia compatibilidad con la distribución de Hadoop y mapR de código abierto.

2.2.8.1. Funcionamiento de RHadoop.

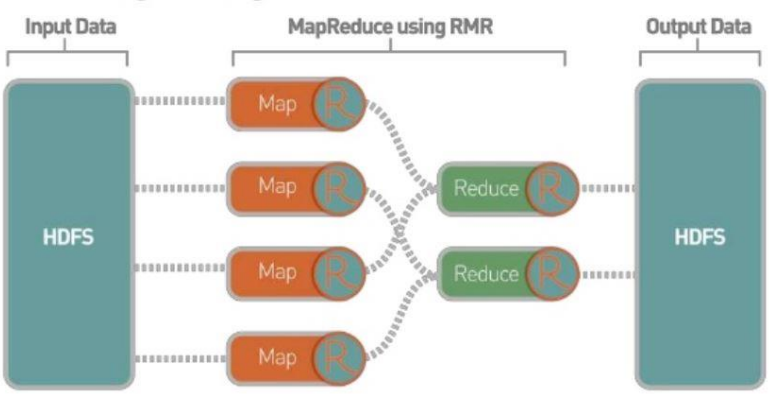


Figura 13 - Funcionamiento de RHadoop
Fuente: Revolutions Analytics
Elaboración: Revolutions Analytics.

RHadoop funciona primeramente realizando el llamado a las librerías rhdfs que es el HDFS y rmr2 que es el MapReduce de Hadoop. Una vez iniciadas las librerías se procede a cargar los valores de la data al HDFS, posterior se procede a realizar el procesamiento de los valores cargados al HDFS mediante MapReduce que es un potente marco de programación para procesar de manera eficiente cantidades muy grandes de datos almacenados en el sistema de archivos distribuido Hadoop. Una vez que se ha realizado el procesamiento de la data el mismo se almacena en el HDFS y es donde hay que recuperarlo para proseguir con el análisis de los mismos y realizar las diferentes operaciones complementarias en R como lo es el filtrado y diseño de cada una de las visualizaciones que se pretenda obtener para comprensión de los mismos, su funcionamiento se representa en la Figura 13.

2.2.8.2. Características y Ventajas de RHadoop.

Entre las características de la utilización de RHadoop se tiene:

- R y Hadoop son un proyecto colaborativo y abierto, la persona que desee trabajar con estas herramientas las puede descargar gratuitamente.
- Con la integración entre R y Hadoop los analistas de Big Data pueden analizar y procesar gran cantidad de datos de una forma sencilla mediante la utilización de los comandos y librerías de R aprovechando el poder del procesamiento de Hadoop.
- Al realizar el análisis del volumen de datos en R, se tiene la ventaja de la utilización de sus métodos de agrupamiento, filtrado y capacidad de diseñar gráficas detalladas y precisas que permiten la correcta interpretación de los resultados obtenidos por parte del equipo de analistas.

- Existe gran cantidad de información de cómo utilizar R, su comunidad es muy activa y se puede obtener solución a cualquier problema con el que se pueda encontrar.
- En pocas líneas de comandos se puede completar un trabajo de análisis, procesamiento y visualización de los datos, además de que es una herramienta estable no tiene errores de funcionamiento graves a pesar de ser gratuita.
- Es fácil de aprender y comprender su funcionamiento, una vez que se haya familiarizado en el trabajo de análisis, procesamiento y visualización de los resultados esperados, la experiencia adquirida es de mucha importancia para trabajos futuros.
- R es uno de los lenguajes más utilizados en el ámbito académico a nivel mundial para completar trabajos de análisis de datos y generación de visualizaciones, al ser software libre la comunidad tiene la capacidad de crear y publicar diferentes librerías que permitan trabajar en un sinnúmero de proyectos relacionados al análisis de datos.

Figura 14 - Big Data Analytics
Fuente: Webopedia
Elacoración: Webopedia

2.3. Big Data Analytics.

Los autores Singh & Singh, (2012), definen a Big Data Analytics como "Las técnicas analíticas avanzadas que operan en grandes conjuntos de datos. Por lo tanto, el gran análisis de datos se centra en dos cosas: los grandes datos y la analítica, además de cómo los dos se han unido para crear una de las tendencias más profundas en la inteligencia empresarial (BI) de hoy." La figura 14, representa de forma global a todos los términos que están relacionados a Big Data Analytics.

Ames, Abbey, & Thompson, (2013), definen a Big Data Analytics como el "proceso de examinar gran cantidad de datos para descubrir patrones ocultos, correlaciones desconocidas e información útil. Tal información proporciona ventajas competitivas a través de organizaciones rivales y resultar en beneficios para el negocio, como el marketing más efectivo y mayores ingresos".

Por lo tanto, con las definiciones anteriores se puede determinar que Big Data Analytics es el conjunto de utilización de procesos y técnicas, con los cuales se puede examinar gran cantidad de datos, los que a su vez permiten analizarlos y poder entenderlos para posteriormente aplicar alguna regla de negocio y sacarles provecho a estos datos.

2.3.1. Historia y Evolución.

La Historia de Big Data Analytics va de la mano con la historia de Big Data, mientras Big Data trata del aparición y generación de datos, Big Data Analytics trata del manejo e interpretación de estos datos, por eso a continuación se tratará una breve historia como se empezó a utilizar y evolucionar Big Data Analytics basado en un artículo de Zakir, Seymour, & Berg, (2015).

Todo comenzó con el surgimiento de procesar y resguardar los datos generados del Censo de Estados Unidos en el año de 1880, anunciar los resultados tardó más de 7 años, entonces para el censo de 1890, aparece la máquina de tabular desarrollada por

Herman Hollerith, la cual permitió procesar datos que estaban grabados en tarjetas perforadas.

Con el surgimiento de las computadoras los datos tomaron más importancia, para la década de 1980 aparecen las bases de datos relacionales, con esto surge el lenguaje SQL que sirve para recuperar información de las bases de datos relacionales. Para finales de 1980, los datos continuaron generándose a gran velocidad por los bajos costos de almacenamiento de discos duros, en ese momento William H. Inmon propone la creación y la utilización de un almacén de datos, esta nueva tecnología se diferencia de las bases de datos relaciones por la optimización del tiempo de respuesta a las consultas, todo esto conlleva a la aparición de un nuevo concepto el cual es Inteligencia de Negocios o por sus siglas en ingles BI (Business Intelligence), término propuesto por Howard Dresner en el año de 1989.

Para la década de 1990 aparece el concepto de minería de datos, conocido como el proceso computacional de descubrir patrones en un conjunto de datos para poder analizarlos de una forma diferente a todos los métodos habituales. Esto es posible gracias a las tecnologías de bases de datos y almacenes de datos, los cuales permiten a las empresas almacenar gran cantidad de datos y analizarlos de manera favorable y razonable, ya que solamente están buscando datos específicos, esto a su vez conlleva a predecir patrones de negocio con el cual pueden generar cadenas de negocio específicas.

Continuando con la evolución, manejo y análisis de la información y las tecnologías, aparece el Internet, dada la demanda de la búsqueda de información, noticias, reportes y páginas web, se desarrolla el motor de búsqueda de Google, desarrollado por Larry Page y Sergey Brin, el cual es una tecnología que procesa y analiza gran cantidad de datos en un sin número de computadores y servidores distribuidos. Para principios del 2010 las empresas Amazon y Google, liberan al mercado dos herramientas, una llamada Amazon Redshift, la cual es un almacén de datos en la nube, y Google BigQuery, herramienta que procesa una consulta en cientos de servidores de google, estas herramientas salieron al mercado con bajo costo para que sea accesible a todas las empresas y con la ventaja de procesar grandes cantidades de datos con un presupuesto menor.

Es así que gracias a la historia de la información como a la evolución de Big Data Analytics, hoy en día es posible manejar gran cantidad de datos a bajo costo y con gran eficiencia por las herramientas que se han ido desarrollando como

evolucionando, esto ha ayudado a cada una de las empresas a poder generar estrategias de mercado gracias al estudio y análisis de sus bases de datos.

2.3.2. Ejemplos de la Aplicación de Big Data Analytics

Hoy en día, el término de Big Data y Big Data Analytics están en moda en el entorno empresarial, de acuerdo a una publicación realizada en por Hu, Wen, & Li, (2014), se nombran algunas áreas en las cuales se han aplicado Big Data Analytics, entre las más importantes se tiene:

Tabla 8 Áreas de Aplicación de Big Data Analytics

Área	Descripción
Entendiendo y Segmentando a los Clientes	El objetivo principal es crear modelos predictivos como por ejemplo empresas de supermercados han predicho qué productos se venderán mejor, otro ejemplo es el de las aseguradoras de autos que pueden comprender mejor cómo conducen sus clientes. Incluso las campañas electorales pueden optimizarse gracias a Big Data Analytics, se ha detectado que en la última campaña de Barack Obama se utilizó datos obtenidos para poder analizar y crear un discurso que pueda llegar a la mayoría de votantes.
Entendiendo y Optimizando los Procesos de Negocio	Otra área en la que se ha aplicado Big Data Analytics es en los negocios, ya que gracias a esto están adecuando su stock basándose en predicciones generadas gracias a datos de redes sociales un aspecto el que más ha aprovechado esto son los distribuidores de suministros, los cuales han optimizado las rutas de reparto, gracias a su ubicación geográfica y el monitoreo del trágico en tiempo real. Un claro ejemplo del Big Data Analytics es la película Moneyball que, gracias a la medición estadística, se formó un equipo campeón.
Mejorando la Salud Pública	Se han empleado técnicas de Big Data por ejemplo para monitorizar bebés en la unidad de neonatos de un hospital en Toronto. Grabando y analizando latidos y el patrón de respiración de cada bebé, la unidad ha desarrollado unos algoritmos que pueden predecir infecciones 24 horas antes de que los primeros síntomas aparezcan. De esta manera, el equipo médico puede intervenir y salvar vidas en un entorno en el que cada hora cuenta.

Mejorando el Rendimiento Deportivo	La mayoría de deportistas de élite están adoptando técnicas de Big Data Analytics. Un claro ejemplo es en el tenis, en donde se lleva mucho tiempo utilizando la herramienta SlamTracker, en los torneos más prestigiosos de esta rama, esta plataforma ha logrado registrar datos de más de 8 años de torneos de Grand Slam, en donde se ha podido determinar patrones y estilos de juegos de los ganadores de cada uno de estos torneos.
Mejorando la Ciencia y la Investigación	Otra de las áreas donde más se ha utilizado Big Data Analytics es el área de la ciencia e investigación, en donde el CERN (laboratorio suizo de física nuclear con su gran colisionador de hadrones), uno de los mayores generadores de datos, intenta descubrir los secretos del universo gracias a los datos del acelerador de partículas. Aunque el centro de datos del CERN cuenta con 65.000 procesadores para analizar los 30 petabytes de datos, no es suficiente. Por ello distribuyen la capacidad de computación entre miles de ordenadores repartidos entre otros 150 centros de datos por todo el mundo para analizar los datos.
Optimizando el Rendimiento de Máquinas y Dispositivos	Big Data está ayudando a que la tecnología y las maquinas industriales que se utilizan hoy en día sean autónomas e inteligentes, como por ejemplo Google ha desarrollado un auto que se conduce solo, este auto se encuentra equipado con cámaras, GPS, computadoras con acceso a internet, sensores que permiten al vehículo circular de forma segura por la vía pública sin necesidad de intervención humana.
Comercio Financiero	Las actividades con mayor uso de Big Data son las relacionadas a High-Frequency Trading (HFT), que consiste en una serie de algoritmos para la toma de decisiones de compra venta de valores, teniendo en cuenta además de las señales tradicionales que tienen los comerciantes humanos como el análisis técnico, resultados de empresas, noticias en tiempo real, mensajes de redes sociales, foros, declaraciones públicas de personalidades, etc. Todo esto quiere decir la aparición de un nuevo tipo de datos que anteriormente eran imposible de manejar.

Fuente y Elaboración Propia.

2.3.3. Importancia Big Data Analytics.

Según Sathi, (2012), Big Data Analytics “permite a los analistas, investigadores y usuarios de negocios mejorar la toma de decisiones utilizando datos que antes eran inaccesibles o inutilizables. El uso de técnicas avanzadas de análisis, tales como análisis de texto, aprendizaje automático, análisis predictivo, minería de datos, estadísticas y procesamiento del lenguaje natural, las empresas pueden analizar fuentes de datos sin explotar independiente o junto con sus datos empresariales existentes para obtener nuevos conocimientos que resulta en mucho mejor y más rápido decisiones.”

Big Data Analytics es muy importante para las empresas e instituciones, ya que les permite conocer y aplicar diferentes tipos de estrategias para su beneficio, es así que seguidamente se presenta el resultado de una encuesta realizada por la empresa TWDI en la que pregunto: "¿Cuáles de los siguientes beneficios se obtendrían si tu organización implementaba alguna forma de Big Data Analytics?".

Para mejor comprensión de los resultados se detallará una imagen a continuación, en las cuales se puede identificar las características que se acoplan a Big Data Analytics. En la opción referente a otros beneficios se tiene en cuenta la lealtad de los clientes, la optimización de la experiencia de servicio, la optimización de la prestación de atención sanitaria y el rendimiento del proveedor basado en el coste y la calidad, todos los resultados detallados anteriormente se presentan en la figura 15.

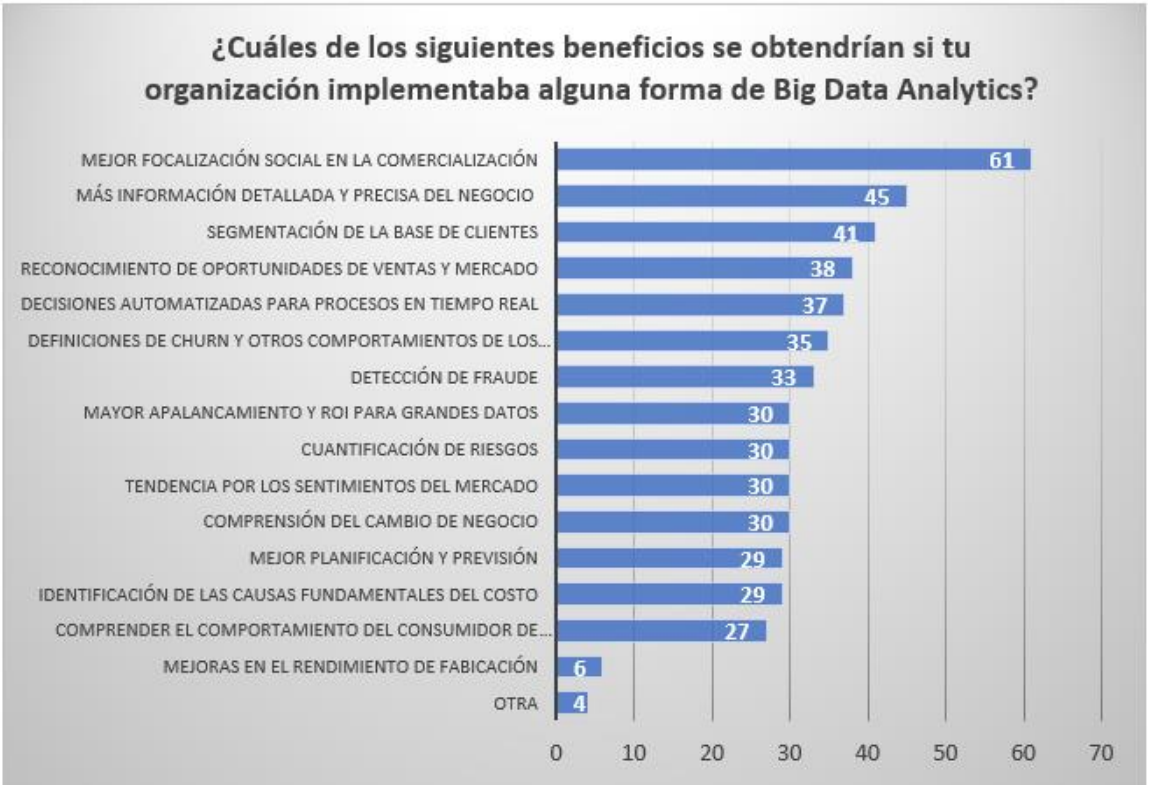


Figura 15 - Encuesta Beneficios Big Data Analytics
Fuente: Autor.
Elaboración: Autor.

Hoy en día el mercado de Big Data Analytics, esta aun en sus primeros pasos, grandes empresas de software como AG, Oracle, IBM, Microsoft, SAP, EMC y HP compiten con otras empresas que se encargan de mantener los datos en la nube.

Se espera con el pasar de los años esta nueva tecnología siga en auge, ya que se ha demostrado que el estudio de Big Data Analytics a partir del análisis de patrones encontrados en los diferentes tipos de Big Data, han abierto la puerta a nuevos estudios de mercado, por parte de los resultados inesperados, que antes no se obtenían porque se aplicaban métodos de presentación de resultados tradicionales.

2.3.4. Ventajas.

Las empresas emplean Big Data Analytics porque buscan encontrar información muy importante y clara en sus datos, es así como muchos proyectos se han desarrollado por la necesidad de responder a las nuevas reglas de negocio. Si una empresa emplea las herramientas y las plataformas correctas para extraer la información requerida, los resultados que se pueden presentar es el aumento en ventas, aumenta la eficiencia y eficacia de las operaciones, servicio, cliente y la gestión de riesgo.

La revista especializada en negocios LOGICALIS, presenta un artículo llamado “En qué consiste Big Data Analytics y cómo beneficia a tu empresa,” (2016), presenta 5

ventajas que ofrece Big Data Analytics para su negocio, las cuales se presentan en la figura 16.

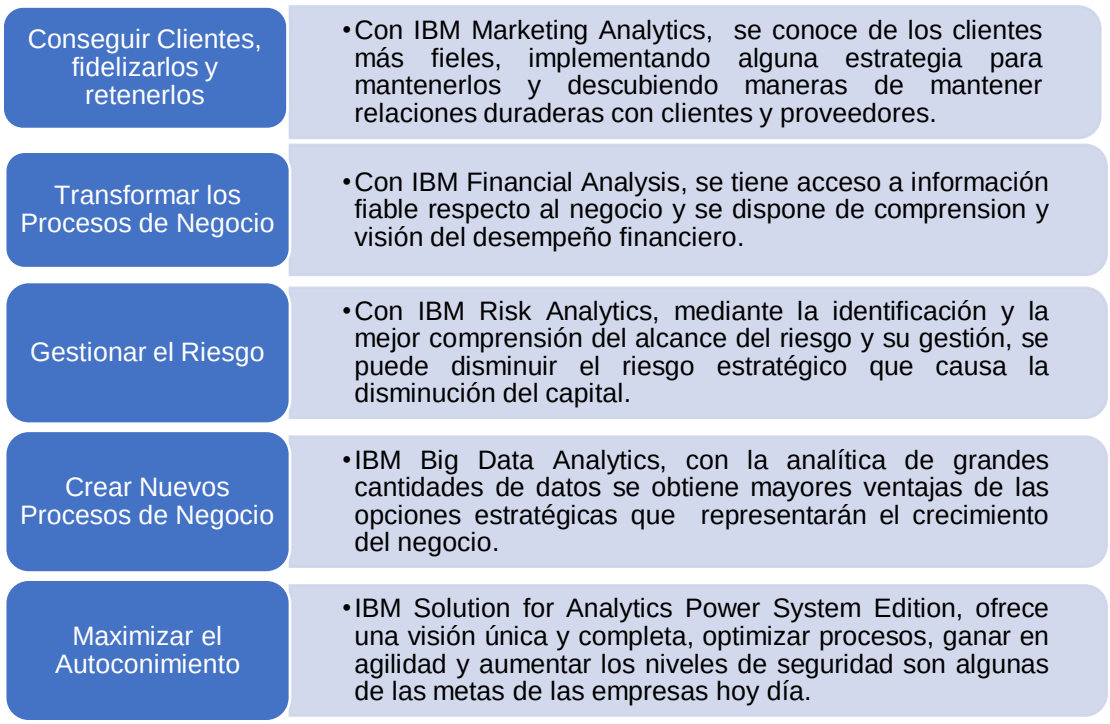


Figura 16 - Beneficios Big Data Analytics
Fuente: Autor
Elaboración: Autor

La revista de tecnología SAS, (2015), realizó un estudio de Big Data Analytics, en el cual pudo determinar otras ventajas que presentan, las cuales se presentan en la figura 17:

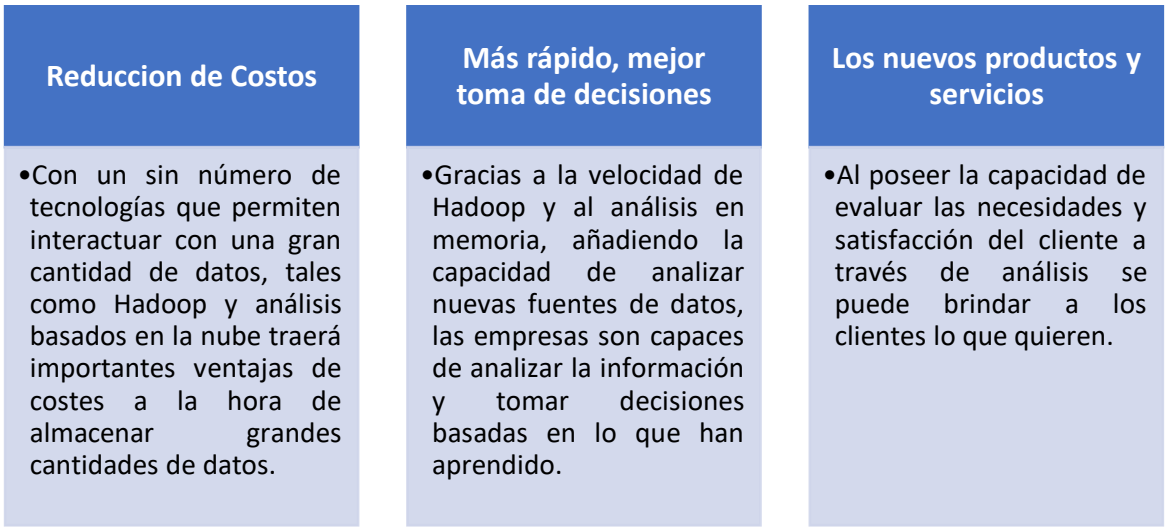


Figura 17 - Beneficios Big Data Analytics
Fuente: Autor
Elaboración: Autor

Con lo mencionado anteriormente se determina que Big Data Analytics es de mucha importancia en el ámbito empresarial, educativo, financiero, salud, deportivo, entre otros. Es por eso que en el mundo digitalizado de hoy es necesario que exista el estudio de los datos generados por la misma empresa y por agentes externos, ya que estos permiten conocer el estado actual de un ámbito estudiado o por estudiar y como fin se puede diseñar o crear nuevas estrategias de negocio que permitan que la empresa obtenga valor determinante para su sustento y desarrollo diario.

Figura 18 - Big Data Toma de Decisiones
Fuente: Trecebits
Elaboración: Trecebits

2.4. Toma Decisiones con Big Data

Mallinger, (2015), menciona que “La toma de decisiones con Big data es muy importante, ya que permite identificar patrones del comportamiento de los clientes, tendencias que haya podido tener un determinado producto, ideas para el desarrollo de un nuevo servicio o producto, etc.”

Para realizar la correcta toma de decisiones no se tiene un tiempo establecido, ya que el tiempo que se dispone para el análisis de los datos puede darse de minutos a días, depende de los tipos de datos que se posea y del análisis de los mismos para la posterior toma de decisiones. Expertos de IBM afirman que, para América Latina, Big Data es una gran oportunidad para que las empresas cambien su forma en la que generan y procesan la información, con la cual podrán enfrentar los problemas y encontrar posibles soluciones eficientes cuando se les presente un problema.

Es así como los métodos de recolección y manejo de información fueron evolucionando a través del tiempo, en la actualidad se posee datos no estructurados y estructurados. Big Data y la toma de decisiones hoy en día es de gran ayuda, ya que nos permite tener una base más sólida donde fundamentar nuestras decisiones.

Las ventajas de la utilización de Big Data son innumerables, pero también debemos reconocer que las desventajas son muchas, algunos ejemplos de la no utilización de Big Data y la toma de decisiones que causaron consecuencias graves son:

- La empresa Kodak, desarrollo la cámara digital en 1975, uno de sus errores fue que al no invertir en tecnología por miedo a que caerían sus ventas de películas, pero sus directivos no previeron la decadencia de la película en la década de los 90. Este error de previsión por parte de sus directivos causo que sus competidores ganaran mercado y Kodak quedara en pérdida de mercado y rentabilidad.

- PRAKTIKER, es una cadena de supermercados alemana, era una empresa modelo, con gran rentabilidad, grandes ingresos y beneficios, pero cuando llego la crisis tomaron una decisión equivocada, la empresa decidió rebajar sus precios a un 20%, al no poseer un sistema que le informara a detalle de sus pérdidas de rentabilidad y un estudio de mercado que les permitiera determinar el tiempo que debían mantener esa oferta, esto produjo sus ingresos cayeran, pero se dieron cuenta demasiadamente tarde.

Como estos casos hay muchos, en los cuales nos damos cuenta que, el análisis de los datos influye en la correcta toma de decisiones o la corrección de los mismos. Es por eso que cada día las empresas requieren de herramientas y conocimiento que les permitan procesar tareas de reportes, para poder utilizar esos datos en estrategias de mercado que ayuden a la empresa a seguir aumentando sus ingresos y mejorar el proceso de toma de decisiones.

Los sectores que han invertido en tecnologías de Big Data y la toma de decisiones son el sector industrial, comercial, la salud, la información, el sector bancario y financiero, instituciones públicas y privadas, etc.

2.5. Trabajos Relacionados.

Existen algunas propuestas de trabajos realizados que permiten el estudio y análisis de Big Data porque se ha convertido en una amplia herramienta para las empresas que les sirve para explotar la riqueza de sus datos. Como resultado, están surgiendo varias soluciones y herramientas que permiten el estudio de los datos y poder obtener resultados que permite satisfacer las necesidades de los interesados. Entre algunos trabajos relacionados que se encuentran en la IEEE sobre el estudio de Big Data se tiene los siguientes:

2.5.1. Análisis de data médica e informática del área de salud utilizando Big Data.

Esta propuesta utiliza la herramienta RStudio para realizar el análisis de datos biomédicos que tiene como objetivo facilitar la interpretación, validación, uso y reutilización de conjuntos de datos, centrándose en la publicación de conjuntos de datos biomédicos sobre la hepatitis, estos datos pueden servir como fuente de simulación y modelado computacional relacionadas a esta enfermedad.

Utilizando RStudio y la librería rpart para realizar la clasificación, predicción y generación de árboles de decisión se obtiene la generación de resultados del análisis comparando los resultados de los datos de entrenamiento y prueba. Con este método de exploración y comparación se puede obtener la visión principal de los resultados

que son producidos por la ciencia médica. Los resultados óptimos sirven como referencia para la generación futura y se pueden realizar más mejoras en la interpretación de los datos gracias a esta tecnología.

2.5.2. Minería de Datos basada en la nube mediante la herramienta R.

Esta propuesta trata del estudio y análisis de los datos generados por diferentes redes sociales que se encuentran almacenados en la nube y pueden ser utilizados gracias a la librería RPubs desarrollada por R. La utilización de esta librería permite aprovechar igualmente las ventajas de las "3V (volumen, velocidad y variedad)" del Big Data. Además, con la clasificación de los datos se implementa el algoritmo K-Means para trabajar con un porcentaje de los datos y poder realizar los procesos a una mayor velocidad, permitiendo obtener resultados óptimos y predictivos.

Existen gran cantidad de empresas multinacionales que utilizan R para el estudio de sus datos y transformarlos en negocios más efectivos, Facebook utiliza R para analizar la actualización del estado de Facebook y los gráficos de las redes sociales de Facebook. Google usa R para calcular el ROI en una campaña publicitaria, predecir la actividad económica y hacer que la publicidad en línea sea más efectiva. John Deere usa R para el modelado de series de tiempo y análisis geoespacial. Las empresas anteriormente nombradas utilizan la librería rmr de R que proporciona el multiprocesamiento de MapReduce de los datos de Hadoop, en donde se define que MapReduce es un modelo de programación para expresar cálculos distribuidos sobre cantidades masivas de datos y un marco de ejecución para el procesamiento de datos a gran escala en diferentes máquinas.

2.5.3. Medición inteligente de datos generados por sensores usando R y Hadoop.

En este trabajo relacionado se tiene como punto de partida la implementación de políticas eficientes de conservación de energía para reducir el consumo residencial de electricidad en países europeos y la aparición de medidores eléctricos inteligentes ha abierto el camino para utilizar los datos de utilización de la electricidad. La gran cantidad de datos generados por los medidores inteligentes en cada intervalo se puede utilizar para el análisis de datos y se pueden derivar diversos conocimientos, como prever la demanda de electricidad, implementar tasas arancelarias (tiempo de uso) etc. que ayudarán tanto a las empresas de servicios públicos como a residentes.

En este trabajo se utiliza Hadoop y R integrados, para realizar el consumo de los datos del sistema de archivo distribuido que ofrece Hadoop y es en donde se encuentra almacenada toda la data con la que se puede trabajar, R lo que realiza es el consumo

de una parte de la data con la cual puede trabajar sin miedo a dañar o modificar algún valor y generar modelos de consumo de energía de acuerdo a varios resultados como consumo diario, semanal, mensual y trimestral.

2.6. Análisis de Trabajos Relacionados.

Como referencia al análisis de los trabajos relacionados y la investigación desarrollada, se determina el ambiente actual que se tiene del estudio de Big Data y la herramienta más utilizada como lo es R para obtener resultados gráficos del análisis de los datos.

En la Tabla 9 se puede observar un resumen con las principales características que presentan los trabajos relacionados estudiados brevemente en el apartado anterior, esto con el objetivo de analizar e identificar las características y aspectos en estas propuestas, con respecto a proponer una mejor propuesta.

Tabla 9 Comparación Trabajos Relacionados

Trabajo	Características
Análisis de data médica e informática del área de salud utilizando Big Data.	• Estudio de data médica que sirve para analizar y validar información relevante al estudio de casos clínicos. • Utilización de RStudio juntamente con la librería rpart para crear arboles de decisión. • Generación de resultados gráficos de árboles de decisión para predecir resultados fututos. • Utiliza la librería rJava y rhdfs para integrar con Hadoop y poder consumir la Big Data.
Minería de Datos basada en la nube mediante la herramienta R.	• Estudio de datos de diferentes dominios como comercio electrónico, gestión del tráfico o ciudades inteligentes. • Utilización del lenguaje de programación R y su librería RPubs que permite el estudio y análisis de datos en la nube. • Se empieza a trabajar con un volumen de datos sumamente alto que ya se conoce como Big Data. • Utiliza la librería rmr para trabajar con el multiprocesamiento de la data en Hadoop.
Medición inteligente de datos generados por sensores usando R y Hadoop.	• Estudio de data generada por sensores de consumo de energía. • Primer trabajo relacionado a la utilización del lenguaje de programación R juntamente integrado con Hadoop. • Se aprovecha la utilización de la librería rhdfs para consumir la data almacenada en el HDFS de Hadoop. • Se genera modelos estadísticos de predicción de consumo de energía.

Fuente y Elaboración Propia.

Este trabajo de titulación tiene la finalidad de adquirir la experiencia y la forma de trabajar de cada uno de los trabajos relacionados y gracias al poder que ofrece Hadoop y su funcionalidad como lo es el sistema de archivos distribuido o HDFS y MapReduce que ofrece el multiprocesamiento de los datos, así mismo logrando la integración con la herramienta R gracias a las librerías que otorgan la funcionalidad de Hadoop dentro de R, es posible trabajar con los datos sin tener la necesidad de dañar o modificar la estructura de la misma, una vez que se haya cumplido con la integración, funcionalidad, procesamiento y obtención de los resultados se procede al desarrollo de un prototipo en PHP que permite la ejecución del análisis de los datos analizados y scripts diseñados en RHadoop así mismo la obtención de gráficas con las librerias plot y plotrix que permiten la generación de gráficas de barras y de un pastel en 3D de acuerdo al análisis y procesamiento de los datos trabajados.

2.7. Metodologías Aplicables.

En este apartado se describen 3 metodologías que pueden ser utilizadas para la continuación del desarrollo del trabajo de fin de titulación, estas metodologías permiten el correcto análisis, procesamiento y desarrollo de las pruebas con los datos obtenidos, a continuación, se describe cada una de las mismas.

2.7.1. KDD (Knowledge Discovery in Databases) (Descubrimiento de Conocimientos en Bases de Datos)

Shafique & (Qaiser, 2014), definen a KDD como “El proceso de extracción de los conocimientos ocultos de acuerdo con un volumen de datos. KDD requiere un conocimiento previo relevante y una comprensión breve del dominio y las metas de la aplicación”. Además, se conoce que este proceso no es automático y extrae información de alta calidad la que se puede utilizar para tomar decisiones acertadas en relación con patrones o modelos encontrados dentro del volumen de datos. Las fases con las que cuenta esta metodología se representan en la siguiente figura:

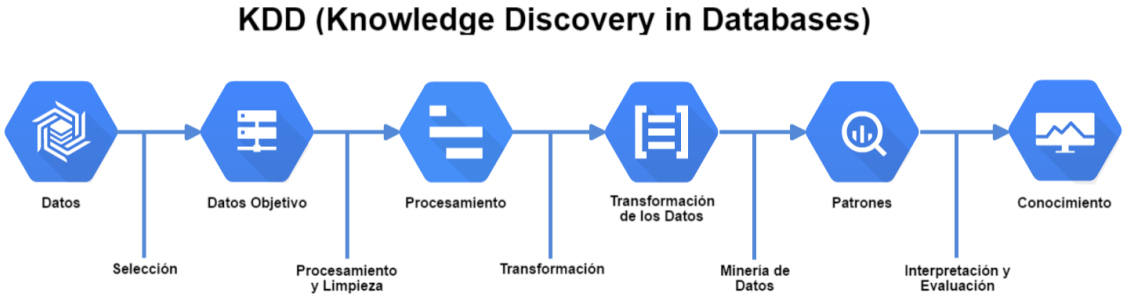


Figura 19 - Fases Metodología KDD
Fuente: Autor
Elaboración: Autor

Para mejor comprensión de esta metodología se explica cada una de sus fases:

2.7.1.1. Selección.

Esta fase consiste en definir los objetivos y las herramientas a utilizar en el proceso de analizar e identificar los datos que se han obtenido, además se identificará sus atributos de entrada y la información que se esperará obtener, en otras palabras, primeramente, se debe conocer lo que se quiere obtener y cuál es el volumen de datos que nos facilitará obtener estos resultados.

2.7.1.2. Limpieza de datos.

En esta fase lo que se realiza la limpieza de datos, incluye la tarea de completar los atributos de los datos que se encuentran incompletos, valores incorrectos e inconsistentes. En algunos casos estos datos deben de ser eliminados ya que sus atributos pueden contribuir a la lectura, análisis y resultados de datos incorrectos.

2.7.1.3. Procesamiento e integración de datos.

Esta fase tiene como fin combinar los datos de múltiple origen, esto incluye múltiples volúmenes o bases de datos, esta información podría tener diferentes formatos y atributos.

2.7.1.4. Transformación de datos.

Esta tarea consiste en la modificación de atributos o datos sin que supongan un cambio en la estructura de la misma, esta transformación de la información tiene la ventaja de mejorar la comprensión de los datos ya que pasan de bajo nivel a alto nivel, lo que al final conlleva que bajen los tiempos de ejecución de los algoritmos de búsqueda, pero, su principal desventaja es que se puede reducir la exactitud del conocimiento descubierto, por causa de la perdida de algún tipo de información.

2.7.1.5. Minería de Datos.

Consiste en la búsqueda e identificación de patrones que puedan utilizar e identificar un modelo que exprese la dependencia de los datos. Se debe tener conocimiento de los datos como que se espera obtener para posterior seleccionar solamente uno de los posibles modelos. Como aspecto adicional se debe especificar una estrategia de búsqueda que se va a utilizar, esto normalmente se encuentra determinado en el algoritmo a utilizar.

2.7.1.6. Evaluación de los patrones.

Esta fase tiene como fin la identificación de los patrones encontrados, pero con la característica de que sean los que verdaderamente ofrecen los resultados esperados e

interesantes, lo que a su vez permitirá representar el conocimiento utilizando técnicas de análisis estadísticos y lenguajes de consultas.

2.7.1.7. Conocimiento e Interpretación de resultados.

Consiste en comprender, analizar he interpretar el conocimiento obtenido de los resultados obtenidos, para mejor comprensión de los mismos puede que sea necesario volver a pasos anteriores.

2.7.2. SEMMA (Sample, Explore, Modify, Model, Assess) (Muestreo, Exploración, Modificación, Modelado, Valoración)

Corrales, Ledezma, & Corrales, (2015), detallan que SEMMA fue desarrollado por el Instituto SAS, en la cual se consideran que su ciclo de vida tiene 5 etapas que son: Muestreo, Explorar, Modificar, Modelar y Evaluar. Comenzando con una muestra estadísticamente representativa de sus datos, SEMMA pretende facilitar la aplicación de técnicas exploratorias estadísticas y de visualización, seleccionar y transformar las variables predictivas más significativas, modelar las variables para predecir los resultados y finalmente confirmar la exactitud de un modelo. Las fases por las que se encuentra conformado son:

Figura 20 - Fases SEMMA
Fuente: Autor.
Elaboración: Autor.

La descripción de cada una de sus fases se describe a continuación:

2.6.2.1. Muestreo.

Esta es la primera etapa del proyecto en la cual se realiza la preparación de los datos para proseguir con la exploración, lo más común de esta etapa es la utilización del nodo de partición, Para la realización de pruebas se divide en porcentaje de 70-30, el 70% que servirá para realización de las muestras y el de 30% que servirá para la validación del volumen de datos.

2.6.2.2. Exploración.

Esta etapa se realiza la exploración de los datos, es la parte más complicada en la cual se posee un nodo que ayudará a la realización de la exploración gráfica, y el otro nodo de selección que permitirá la eliminación de datos que no poseen relación con el objetivo buscado.

2.6.2.3. Modificación.

En esta etapa se realiza la selección y manipulación de los datos para que posean un formato adecuado. El objetivo principal de esta fase es establecer una relación entre las variables explicativas y objeto del estudio, lo que posibilitará deducir el valor de las mismas con un alto nivel de confianza.

2.6.2.4. Modelado.

Esta etapa comprende la selección de los modelos, los cuales dependerán de la cantidad de variables y datos que se posea. Se podrá elegir entre regresión, regresión logística, árboles de decisión, análisis factorial discriminante, redes neuronales, etc. La ventaja que se tiene es que se puede aplicar más de un modelo a la vez, lo que permitirá la comparación de los resultados.

2.6.2.5. Valoración.

Esta etapa comprende la comparación de los modelos una vez ya realizado. Lo que más se utiliza para comparar estos resultados es la utilización del diagrama ROC, este diagrama permite realizar la comparación del comportamiento total del modelo, esta grafica presenta dos variables: la sensibilidad y la especificidad. Lo ideal es que ambas categorías sean altas.

2.7.3. CRISP-DM (Cross Industry Standard Process for Data Mining) (Proceso Estándar Transversal de la Industria para la Minería de Datos)

Yun, Weihua, & Yang, (2014), describen a CRISP-DM como un modelo de proceso jerárquico, que consiste en conjuntos de tareas descritas en varios niveles de abstracción. Esta metodología hace que los grandes proyectos de minería de datos sean más rápidos, más baratos, más confiables y más manejables. El ciclo de vida de esta metodología es:

Figura 21 - Fases CRISP-DM
Fuente: Singular – Data&Analytics.
Elaboración: Singular – Data&Analytics.

2.6.3.1. Definición de necesidades del cliente (comprensión del negocio). (Business Understanding).

Esta fase inicial se centra en la comprensión de los objetivos del proyecto y los requisitos desde una perspectiva de negocio, a continuación, se convierte este conocimiento en una definición de problema de minería de datos y un plan preliminar diseñado para lograr los objetivos.

2.6.3.2. Estudio y comprensión de los datos. (Data Understanding).

La fase de comprensión de los datos comienza con la recolección inicial de datos y prosigue con actividades que le permiten familiarizarse con los datos, identificar problemas de calidad de los datos, descubrir primeros conocimientos sobre los datos y/o detectar subconjuntos interesantes para formar hipótesis sobre información oculta.

2.6.3.3. Análisis de los datos y selección de características. (Data Preparation).

La fase de preparación de datos abarca todas las actividades necesarias para construir el conjunto de datos final (datos que se introducirán en las herramientas de modelado) a partir de los datos iniciales. Es probable que las tareas de preparación de datos se realicen varias veces y no en ningún orden prescrito. Las tareas incluyen tabla, registro y selección de atributo, así como transformación y limpieza de datos para herramientas de modelado.

2.6.3.4. Modelado. (Modeling).

En esta fase, se seleccionan y aplican varias técnicas de modelado, y sus parámetros se calibran a valores óptimos. Normalmente, existen varias técnicas para el mismo tipo

de problema de minería de datos. Algunas técnicas tienen requisitos específicos sobre la forma de los datos. Por lo tanto, volver a la fase de preparación de datos es a menudo necesario.

2.6.3.5. Evaluación (obtención de resultados). (Evaluation).

En esta etapa del proyecto, se ha creado un modelo (o modelos) que parecen tener alta calidad desde una perspectiva de análisis de datos. Antes de proceder al despliegue final del modelo, es importante evaluarlo a fondo y revisar los pasos ejecutados para crearlo, para asegurarse de que el modelo cumple adecuadamente los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión comercial importante que no se ha considerado suficientemente. Al final de esta fase, se debe llegar a una decisión sobre el uso de los resultados de la extracción de datos.

2.6.3.6. Despliegue (puesta en producción). (Deployment).

La creación del modelo generalmente no es el final del proyecto. Incluso si el propósito del modelo es aumentar el conocimiento de los datos, el conocimiento adquirido tendrá que ser organizado y presentado de manera que el cliente pueda usarlo.

2.8. Comparación entre KDD, SEMMA y CRISP-DM

En la tabla número 9, se realiza la comparación de las metodologías estudiadas, de acuerdo con sus principales características de funcionamiento, lo cual permitirá conocer y demostrar cual es la más adecuada para realizar las tareas de análisis de los datos y obtención de resultados.

Tabla 10 - Comparación Metodologías

Características	KDD	SEMMA	CRISP-DM
Pre selección de Datos			X
Selección y muestra de Datos	X	X	X
Exploración y Preprocesamiento de Datos	X	X	X
Modificación y Transformación de Datos	X	X	X
Modelado y Minería de Datos	X	X	X
Interpretación y Evaluación	X	X	X
Post análisis y Conocimiento Adquirido			X

Fuente y Elaboración Propia.

Con la tabla número 10, se demostró que la metodología que cumple con las características más completas y permite el correcto desarrollo del proyecto es CRISP-DM en comparación con las otras metodologías mencionadas, esta es la razón por lo cual se ha escogido la metodología CRISP-DM para la continuación de las siguientes

fases del proyecto, por las ventajas y descripciones que presenta cada una de las etapas, además que esta metodología se adapta completamente con los objetivos buscados en el desarrollo de este proyecto como son el análisis, procesamiento, obtención de resultados y toma de decisiones de un gran volumen de datos.

CAPITULO III PROBLEMÁTICA

3.1. Planteamiento y Análisis.

El presente trabajo de fin de titulación tiene como objetivo principal realizar el análisis y visualización gráfica de un Big Data obtenido de un sistema de información. El análisis, exploración y visualización de los datos se realiza con la herramienta R Studio, la cual se encuentra integrada con Hadoop para así llegar a formar RHadoop, estos resultados obtenidos pretenden ser utilizados para realizar toma de decisiones.

Para el desarrollo de este trabajo se plantea implementar una arquitectura multinodo de Hadoop en la cual se realiza la instalación e integración entre R y Hadoop.

3.2. Planteamiento del Problema.

La generación de datos en nuestro día a día ha sobrepasado nuestras expectativas, el estudio y análisis de los mismos se convertirá en una base clave para crear nuevas oportunidades, productividad e innovación de negocio. En el día a día el análisis de los datos es de suma importancia para cada sector ya sea tecnológico, médico o empresarial, ya que sin el mismo no se podría crear nuevas oportunidades que permitan un crecimiento de negocio. En muchos casos estos datos generados no son tomados en cuenta, además existen pocos expertos que puedan realizar un análisis detallado de los mismos, es así como las empresas dejan de lado el estudio de sus datos y en muchos casos fracasan. En nuestra universidad existe gran cantidad de datos estudiantiles generados cada semestre que no son analizados ni se puede generar una visualización que permita una comprensión y análisis de los mismos, así mismo las autoridades competentes no tienen conocimiento de los resultados obtenidos por semestre y así no se realiza un seguimiento adecuado, esto conlleva a que no se pueda realizar una correcta toma de decisiones ni nuevas estrategias de negocio.

Frente a esta problemática en este trabajo se realiza el análisis de dichos datos, mediante la herramienta de análisis R que a su vez se encuentra integrado con Hadoop y aprovechar su poder de procesamiento, con los resultados obtenidos de este análisis se pretende diseñar una serie de visualizaciones que permitan una mejor comprensión de los mismos, y como paso final se pretende el desarrollo de un prototipo en el cual se presentarán todos los resultados de este análisis que puede ser presentado a cualquier autoridad que necesite de los mismos y le permita una correcta toma de decisiones y a su vez crear nuevas estrategias de negocio.

3.3. Justificación.

Los motivos que conllevan al desarrollo de este trabajo son:

1. Analizar y generar visualizaciones mediante RHadoop de estos datos que permitan una mejor comprensión de los mismos y aporten a la toma de decisiones.
2. Que los resultados obtenidos de este análisis y visualizaciones sean de utilidad a los diferentes interesados para una correcta toma de decisiones y permita la creación de nuevas estrategias de negocio para que exista un precedente del análisis y visualización de los mismos.
3. Implementar una arquitectura multinodo de Hadoop y a su vez funcione integrado con R para poder procesar, analizar y obtener resultados de los datos a trabajar.

3.4. Solución Propuesta.

En el presente trabajo se desarrolla una investigación sobre temas relevantes a Big Data y las herramientas a utilizar para realizar el análisis y visualización de los datos obtenidos, así mismo cómo estos resultados pueden ayudar a la correcta toma de decisiones. A partir de la investigación se realiza la configuración de las diferentes herramientas a utilizar como lo son R y Hadoop, así mismo integrar dichas herramientas y permitir que Hadoop funcione multinodo, ya que se ha comprobado el correcto funcionamiento de las herramientas y equipos a utilizar se procede con el estudio y análisis de los datos obtenidos, una vez que se comprende con que datos se está trabajando se procede a diseñar las diferentes estrategias para obtener diferentes tipos de resultados que a su fin conlleva al desarrollo de las diferentes visualizaciones a presentar. Ya que se haya cumplido con el correcto análisis y visualización de los datos se pretende realizar un prototipo donde se presenten los mismos y sirvan para la correcta toma de decisiones.

La arquitectura que se implementa para la solución del problema es la siguiente:

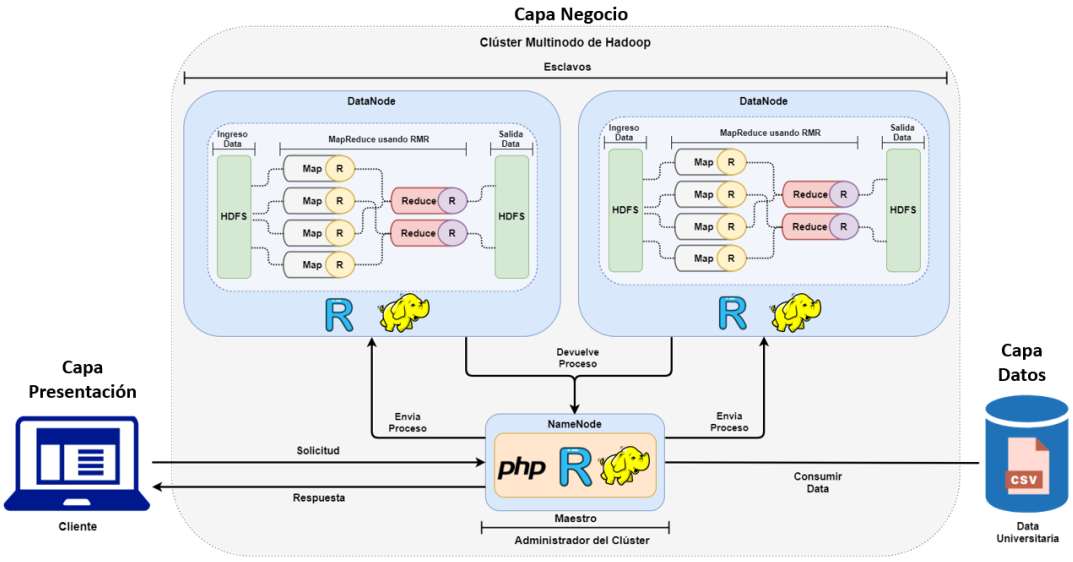


Figura 22 - Arquitectura de la Solución
Fuente: Autor
Elaboración: Autor

Como se puede apreciar en la figura 22, la arquitectura que se implementa en el desarrollo de este trabajo de titulación es una arquitectura 3 capas, a continuación, se describe cada una de sus capas:

- **Capa de Presentación:** En esta capa se encuentra la interfaz de usuario, en donde el usuario final interactúa con la aplicación y realiza las diferentes solicitudes para obtener un resultado final.
- **Capa de Negocio:** En esta capa se realizan todas las operaciones lógicas de la aplicación, en esta capa la arquitectura multimodo de Hadoop envía los procesos a cada uno de sus nodos para cumplir con el procesamiento solicitado de la capa de presentación.
- **Capa de Datos:** En esta capa se encuentra la conexión y lectura del archivo CSV, el cual es consumido por la capa de negocio para procesar la solicitud enviada por el usuario final.

CAPITULO IV DESARROLLO DE LA SOLUCIÓN E IMPLEMENTACION DEL CASO DE ESTUDIO

4.1. Descripción de la Solución.

Para realizar el análisis, procesamiento y exploración de la data se utiliza el lenguaje R que se encuentra integrado con Hadoop, las librerías de Hadoop desarrolladas para la utilización en R son rmr2(MapReduce) y rhdfs(HDFS), mediante el programa R Studio, el cual permite la exploración y diseño de las visualizaciones. Dentro del entorno de R se procederá a realizar el llamado del entorno(enviroment) o CMD de Hadoop, el cual ayuda al procesamiento de la data entre el master(maestro) y los slaves(esclavos), una vez realizado el procesamiento de la data mediante Hadoop se procede a recuperar el resultado del mismo, el cual sirve para realizar la exploración detallada y diseño de las visualizaciones de la data, esto se logra al gran poder de las librerías y funciones entre R y Hadoop.

La arquitectura aplicada para el desarrollo de este trabajo de titulación es la siguiente:

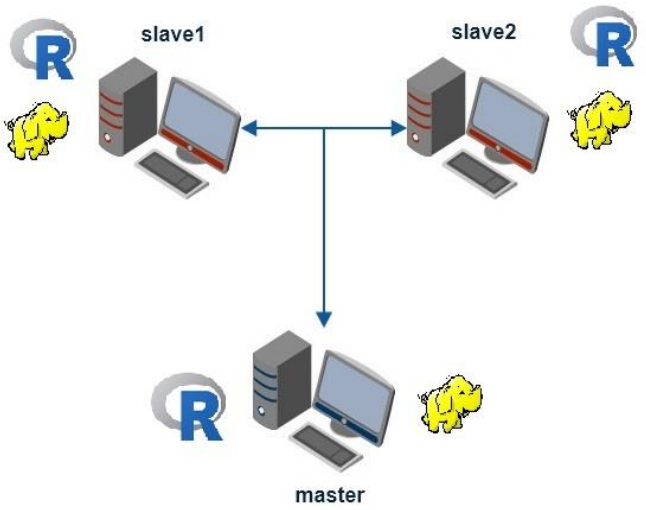


Figura 23 - Infraestructura Diseñada
Fuente: Autor
Elaboración: Autor

Los equipos utilizados tienen instalado el sistema operativo Ubuntu 14.04, además en cada uno se realiza la instalación de Hadoop y R, la versión instalada de Hadoop es 2.6.5 la cual es una versión estable que permitió la integración con R, una vez instalado Hadoop en cada equipo se procede a editar los archivos de configuración para que funcione en forma distribuida, para la realización de este trabajo de titulación se utiliza 3 equipos en los cuales uno es el nodo central o también llamado master, en el que se instala y se integra R, en este nodo master se realiza la exploración y visualización de la data, además se utiliza dos nodos que serán los slaves, en los cuales se debe instalar y configurar R con cada uno de los paquetes proporcionados por Hadoop para su integración. Para una mejor comprensión revisar los 4 primeros

anexos en donde se explica detalladamente la instalación, configuración e integración de Hadoop y R.

4.2. Implementación de la Metodología CRISP-DM.

En esta sección se detalla la parte práctica del trabajo de titulación, en donde se explica la aplicación de cada una de las fases de la metodología CRISP-DM al problema de la exploración y visualización del volumen de datos universitario. Las fases de la metodología CRISP-DM son:

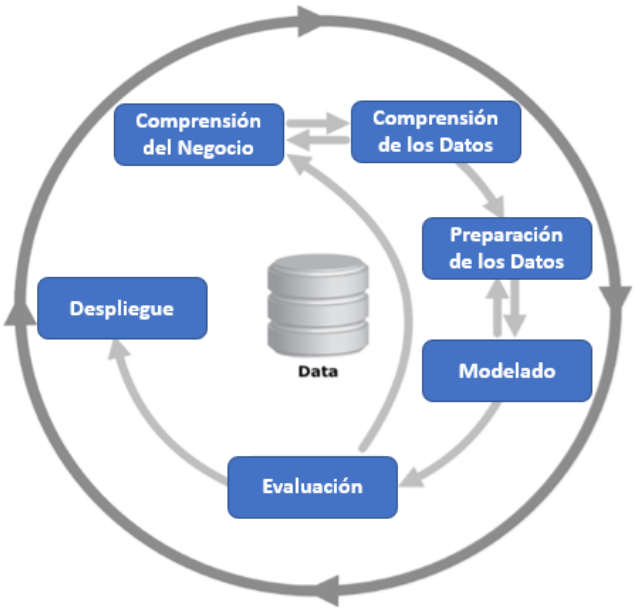


Figura 24 - Fases Metodología CRISP-DM
Fuente: Autor
Elaboración: Autor

4.2.1. Definición de necesidades del cliente (comprensión del negocio).

A continuación, se va explicando cada una de las tareas de esta primera fase en el proceso de la exploración de datos, la cual tiene la finalidad de determinar los objetivos y requisitos del proyecto.

3.2.1.1. Determinar las Necesidades del Cliente.

El objetivo principal de este trabajo de titulación es realizar el análisis, exploración y visualización de los datos bridados y obtenidos por un sistema que posee la universidad. Con los resultados obtenidos se pretende proporcionar una mejor comprensión de los datos, lo cual ayudará a una serie de toma de decisiones, gracias a las visualizaciones obtenidas y diseñadas por Rhadoop.

Contexto.

El principio de este trabajo se realiza por la situación que se cuenta con un volumen de datos generados por uno de los sistemas con los que cuenta la organización, estos

datos contienen información relevante. Sin embargo, no existe ningún estudio a profundidad de estos por lo cual se pretende explorar y obtener visualizaciones que permitan la toma decisiones. Por fines de confidencialidad algunos puntos se obvian en el presente documento, para conocer los detalles que se omiten por favor comunicarse con el autor del documento.

Objetivos del Negocio.

Los objetivos del negocio como se ha mencionado anteriormente es el análisis y visualización de los datos mediante la integración de las herramientas R y Hadoop, las cuales permitan realizar una serie de toma de decisiones. Se podría realizar una serie de visualizaciones de los datos gracias a la exploración minuciosa y estas gráficas pueden ser de mucha utilidad para cada una de las autoridades que componen la organización, ya que permiten conocer e identificar alguna problemática. Esto permitirá a las autoridades competentes a tomar decisiones que ayuden a mejorar la calidad de los servicios brindados.

3.2.1.2. Evaluación de la Situación.

Se cuenta con una data de información detallada, la cual se puede afirmar que contiene una cantidad de datos suficiente para poder realizar la exploración y visualización de la misma.

Inventario de Recursos.

En cuanto a recursos de software para el análisis de los datos se dispone del programa R que se encuentra integrado junto a Hadoop, los cuales juntos proporcionan librerías para realizar el análisis y visualización de los mismos.

En cuanto a hardware se dispone de dos equipos Mac en los cuales se encuentran instaladas 3 máquinas virtuales con el sistema operativo Ubuntu 14.04, las características de los equipos y máquinas virtuales son:

- Equipos MAC.
    - Mac Pro (Late 2013)
    - Procesador 3,5 GHz 6-Core Intel Xeon E5
    - Memoria: 64 GB
- Máquinas Virtuales Ubuntu.
    - Ubuntu 14.04 LTS
    - Procesador 3,5 GHz 6-Core Intel Xeon E5
    - Memoria: 4 GB

Costos y Beneficio.

Los datos utilizados en el presente trabajo no suponen ningún costo adicional a la organización, ya que son datos propios generados por uno de los diferentes sistemas que posee la misma.

El beneficio que puede generar el presente trabajo puede suponer la comprensión y visualización de los datos que permitiría la toma de decisiones para las diferentes autoridades que deseen conocer estos resultados.

3.2.1.3. Determinar los Objetivos de la Minería de Datos.

El objetivo principal de la minería de datos en este proyecto es extraer la información detallada que contiene la data obtenida de uno de los diferentes sistemas con los que cuenta la organización, para así poder realizar un análisis y generación de visualizaciones que permitan la toma de decisiones por parte de las autoridades competentes.

3.2.1.4. Realizar el Plan del Proyecto.

El presente trabajo se dividirá en las siguientes etapas para facilitar su organización y estimar un tiempo de realización:

- Etapa 1: Análisis de la estructura de los datos y la información relevante de la misma. Tiempo estimado: 2 semanas.
- Etapa 2: Ejecución de ejemplos de organización de los datos para su posterior análisis y realización de las visualizaciones. Tiempo estimado: 2 semanas.
- Etapa 3: Preparación de los datos para facilitar el análisis y visualización sobre ellos. Tiempo estimado: 3 semanas.
- Etapa 4: Elección, agrupamiento y generación de los diferentes tipos de visualizaciones de los datos. Tiempo estimado: 1 semana.
- Etapa 5: Analizar los resultados obtenidos en el paso anterior, si fuera necesario repetir la etapa 4 para obtener otros tipos de resultados. Tiempo estimado: 1 semana.
- Etapa 6: Puesta en producción de un prototipo que permita la visualización de los datos para la toma de decisiones. Tiempo estimado: 3 semanas.
- Etapa 7: Presentación del prototipo y los resultados finales. Tiempo estimado: 2 semana.

Evaluación inicial de herramientas y técnicas.

La herramienta que se va a utilizar para cumplir con los objetivos de este trabajo es R Studio, la cual se encuentra integrada con Hadoop, esta herramienta se adapta a la

metodología que se emplea. Además, gracias a esta herramienta no es necesario pasar la información almacenada a una base de datos o a otra herramienta de minería de datos, ya que R Studio opera directamente sobre los datos.

4.2.2. Estudio y comprensión de los datos.

En la segunda fase de la metodología CRISP-DM se realiza la recolección de los datos para poder familiarizarse y reconocer la calidad de los mismos, así como identificar los diferentes tipos de relación y poder establecer las primeras conclusiones.

4.2.2.1. Recolectar los Datos Iniciales.

Los datos que se utilizan para obtener los resultados de análisis y visualización es información general obtenida de los diferentes sistemas de información con los que cuenta la organización. El objetivo principal del estudio de estos datos es el análisis y visualización para la toma de decisiones.

4.2.2.1. Exploración de los Datos.

Una vez que se ha detallado que información contiene cada uno de los atributos, se procede a la exploración de los mismos para comprobar la consistencia e información relevante para poder generar visualizaciones que permitan una mejor comprensión y si las mismas ayudan a la toma de decisiones.

4.2.2.2. Verificar la Calidad de los Datos.

Luego de realizar la exploración inicial de los datos se puede determinar que en la mayoría los mismos se encuentran completos. Los datos obtenidos cumplen con los requerimientos esperados, con los cuales se puede generar las respectivas visualizaciones. Los datos no contienen valores fuera de rango ni errores que influyan en resultados inesperados. Se puede determinar que la data utilizada contiene algunos campos innecesarios los cuales no son tomados en cuenta para su análisis ya que con esos datos se obtiene visualizaciones que no permiten una correcta comprensión ni toma de decisiones.

4.2.3. Análisis, Preparación de los datos y selección de características.

En esta fase se prepara los datos para poder ser analizados profundamente y que permitan que los resultados sean más precisos y las visualizaciones sean más detalladas y entendibles, esto implica seleccionar un pequeño conjunto de datos y trabajar con ellos, para que sean lo más limpios posibles y devuelvan resultados óptimos.

4.2.3.1. Selección de los Datos.

La data que se utiliza tiene aproximadamente más de 500 mil registros, sin embargo, existen algunos campos que no son tomados en cuenta para la realización del análisis y visualización, ya que son campos no necesarios para cumplir con los objetivos planteados.

El motivo por lo cual se excluyen algunos campos es porque no cumplen con la importancia en relación con los objetivos planteados anteriormente.

4.2.3.2. Limpiar los Datos.

La data con la que se cuenta para el estudio de este trabajo contiene toda la información relevante y necesaria para poder cumplir con todos los objetivos planteados, una ventaja adicional que se tiene con la data que se posee es que los datos son limpios y no existe la necesidad de realizar una limpieza sobre ellos.

Para poder realizar el análisis y la visualización de los datos es necesario realizar un filtrado por cada uno de los objetivos planteados, con esto se obtendrá la selección de los datos cuyos campos son los necesarios para realizar su análisis y posterior visualización.

4.2.3.3. Construir los Datos.

En este apartado se destaca el análisis y transformación de los datos con información acerca de los atributos más relevantes para elaboración de un mapa, lo cual permite generar una imagen de un mapa nacional con la información extraída de la data general.

4.2.3.4. Formateo de los Datos.

El campo que contiene la información sobre los datos a nivel nacional tuvo que ser analizado más detenidamente para poder realizar una clasificación por provincia, ya que para la generación de una imagen de un mapa nacional fue necesario trabajar con una librería específica de R y datos obtenidos de la página oficial del INEC, sin realizar lo antes mencionado era imposible trabajar y generar una imagen con dichos datos.

4.2.4. Modelado.

En esta fase se escoge la técnica más apropiadas de agrupación para cumplir los objetivos anteriormente descritos. A continuación, se detalla la aplicación de agrupamiento de los datos para generar los resultados esperados, una vez realizado el plan de pruebas para los datos, se procederá a aplicar las técnicas de agrupamiento que permitan generar un modelo de evaluación y si este ha cumplido los criterios de éxito o no.

4.2.4.1. Escoger la Técnica de Modelado.

Debido que se va a utilizar el software R Studio juntamente integrado con Hadoop para realizar el análisis, exploración y visualización de una data, se utiliza algunas técnicas de agrupación de acuerdo con los objetivos planteados anteriormente en este trabajo.

4.2.4.2. Construir el Modelo.

En este apartado se procede a ejecutar el modelo de agrupamiento elegido sobre los datos de prueba. A continuación, se describen los ajustes de este agrupamiento y el resultado obtenido de este agrupamiento.

Ajustes de parámetros

Ya que se han definido los objetivos, a continuación, se divide esta sección en cuatro partes, una por cada objetivo, ya que los resultados variaran de acuerdo a los parámetros seleccionados.

- Objetivo 1: Identificar los atributos por información general.

```
prespreg.values <- to.dfs(prespreg)
proces <- mapreduce(input=prespreg.values)
datproc <- from.dfs(proces)
presproc <- datproc$val

notp <- table(presproc$                )
```

Figura 25 - Modelo Agrupación Calificaciones
Fuente: Autor
Elaboración: Autor

- Objetivo 2: Identificar el número de atributos anteriores.

```
prespreg.values <- to.dfs(prespreg)
proces <- mapreduce(input=prespreg.values)
datproc <- from.dfs(proces)
dataprc <- datproc$val

deppre <- table(dataprc      )
```

Figura 26 - Modelo Agrupación Titulaciones
Fuente: Autor
Elaboración: Autor

- Objetivo 3: Identificar el estado de un registro de cada campo de la data.

```
prespreg.values <- to.dfs(prespreg)
proces <- mapreduce(input=prespreg.values)
datproc <- from.dfs(proces)
presproc <- datproc$val

deppre <- table(presproc$ESTADO_REGISTRO)
```

Figura 27 - Modelo Agrupamiento Estado de Registro
Fuente: Autor
Elaboración: Autor

- Objetivo 4: Identificar la cantidad de datos a nivel nacional por provincia.

```
dat.values <- to.dfs(dat)
proces <- mapreduce(input=dat.values)
dataprocs <- from.dfs(proces)
datos <- dataprocs$val

matprov <- table(datos
```

Figura 28 - Modelo Agrupamiento Centros
Fuente: Autor
Elaboración: Autor

Modelos

En este apartado se ejecuta cada uno de los modelos de agrupamiento diseñado para los objetivos mencionados anteriormente, primeramente, se realiza la prueba con un 60% de los datos los cuales son los de entrenamiento, dejando el 40% de los datos para realizar un conjunto de pruebas. Los detalles de cada modelo se presentan a continuación:

- Modelo para el objetivo 1.

```
> presencialpreg.values <- to.dfs(presencialpreg)
17/12/20 18:53:39 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
17/12/20 18:53:39 INFO compress.CodecPool: Got brand-new compressor [.deflate]
> proces <- mapreduce(input=presencialpreg.values)
17/12/20 18:54:12 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
packageJobJar: [/tmp/hadoop-unjar1788804977260319212/] [] /tmp/streamjob4502589486226934487.jar tmpDir=null
17/12/20 18:54:13 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/20 18:54:13 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/20 18:54:14 INFO mapred.FileInputFormat: Total input paths to process : 1
17/12/20 18:54:14 INFO mapreduce.JobSubmitter: number of splits:2
17/12/20 18:54:14 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1513807224474_0005
17/12/20 18:54:15 INFO impl.YarnClientImpl: Submitted application application_1513807224474_0005
17/12/20 18:54:15 INFO mapreduce.Job: The url to track the job: http://hadoop:8088/proxy/application_1513807224474_0005/
17/12/20 18:54:15 INFO mapreduce.Job: Running job: job_1513807224474_0005
17/12/20 18:54:20 INFO mapreduce.Job: Job job_1513807224474_0005 running in uber mode : false
17/12/20 18:54:20 INFO mapreduce.Job:  map 0% reduce 0%
17/12/20 18:54:32 INFO mapreduce.Job:  map 21% reduce 0%
17/12/20 18:54:41 INFO mapreduce.Job:  map 100% reduce 100%
17/12/20 18:54:41 INFO mapreduce.Job: Job job_1513807224474_0005 completed successfully
17/12/20 18:54:41 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=223656
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=897876
                HDFS: Number of bytes written=16664198
                HDFS: Number of read operations=14
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=36973
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=36973
                Total vcore-milliseconds taken by all map tasks=36973
                Total megabyte-milliseconds taken by all map tasks=37860352
        Map-Reduce Framework
                Map input records=21
                Map output records=26
                Input split bytes=186
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=102
                CPU time spent (ms)=22840
                Physical memory (bytes) snapshot=503042048
                Virtual memory (bytes) snapshot=2121351168
                Total committed heap usage (bytes)=434110464
        File Input Format Counters
                Bytes Read=897690
        File Output Format Counters
                Bytes Written=16664198
17/12/20 18:54:41 INFO streaming.StreamJob: Output directory: /tmp/filee835f3f500e
17/12/20 18:54:45 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee83b03c129
17/12/20 18:54:49 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee831d80f0a6
> dataprocs <- from.dfs(proces)
17/12/20 18:55:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee83276c9572
> prespreproce <- dataprocs$val
> notpreg <- table(prespreproce$NOTA_FINAL_EVAL)
> notpreg

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26
 253   42   69   51   69   72   77   83   72  106  100   98  107   97  113  109  153  135  168  177  233  207  259  260  299  291  294
  27   28   29   30   31   32   33   34   35   36   37   38   39   40
 131 4409 2289 2474 2159 2312 2150 2183 1784 2020 1424 1639  883 1419
```

Figura 29 - Resultado Modelo Calificaciones
Fuente: Autor
Elaboración: Autor

- Modelo para el objetivo 2.

```
> presencialpreg.values <- to.dfs(presencialpreg)
17/12/18 20:50:19 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
17/12/18 20:50:19 INFO compress.CodecPool: Got brand-new compressor [.deflate]
> proces <- mapreduce(input=presencialpreg.values)
17/12/18 20:50:52 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
packageJobJar: [/tmp/hadoop-unjar8458238291824370108/] [] /tmp/streamjob6795636425802039233.jar tmpDir=null
17/12/18 20:50:52 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/18 20:50:52 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/18 20:50:53 INFO mapred.FileInputFormat: Total input paths to process : 1
17/12/18 20:50:54 INFO mapreduce.JobSubmitter: number of splits:2
17/12/18 20:50:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1513646413556_0006
17/12/18 20:50:54 INFO impl.YarnClientImpl: Submitted application application_1513646413556_0006
17/12/18 20:50:54 INFO mapreduce.Job: The url to track the job: http://hadoop:8088/proxy/application_1513646413556_0006/
17/12/18 20:50:54 INFO mapreduce.Job: Running job: job_1513646413556_0006
17/12/18 20:50:59 INFO mapreduce.Job: Job job_1513646413556_0006 running in uber mode : false
17/12/18 20:50:59 INFO mapreduce.Job:  map 0% reduce 0%
17/12/18 20:51:11 INFO mapreduce.Job:  map 21% reduce 0%
17/12/18 20:51:20 INFO mapreduce.Job:  map 100% reduce 100%
17/12/18 20:51:20 INFO mapreduce.Job: Job job_1513646413556_0006 completed successfully
17/12/18 20:51:20 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=223654
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=897874
                HDFS: Number of bytes written=16664198
                HDFS: Number of read operations=14
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=37383
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=37383
                Total vcore-milliseconds taken by all map tasks=37383
                Total megabyte-milliseconds taken by all map tasks=38280192
        Map-Reduce Framework
                Map input records=21
                Map output records=26
                Input split bytes=184
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=187
                CPU time spent (ms)=23100
                Physical memory (bytes) snapshot=437534720
                Virtual memory (bytes) snapshot=2108727296
                Total committed heap usage (bytes)=409468928
        File Input Format Counters
                Bytes Read=897690
        File Output Format Counters
                Bytes Written=16664198
17/12/18 20:51:20 INFO streaming.StreamJob: Output directory: /tmp/filee3c1f4df5cb
17/12/18 20:51:24 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee3c39b45d61
17/12/18 20:51:28 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee3c493da6e0
> dataprocs <- from.dfs(proces)
> matrpreproce <- dataprocs$val
> deppre <- table(matrpreproce$AREA)
> deppre1 <- deppre[deppre>0]
> deppre1
```

Figura 30 - Resultado Modelo Áreas
Fuente: Autor
Elaboración: Autor

- Modelo para el objetivo 3.

```
> presencialpreg.values <- to.dfs(presencialpreg)
17/12/18 20:55:07 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
17/12/18 20:55:07 INFO compress.CodecPool: Got brand-new compressor [.deflate]
> proces <- mapreduce(input=presencialpreg.values)
17/12/18 20:55:48 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
packageJobJar: [/tmp/hadoop-unjar9157950537771704474/] [] /tmp/streamjob7599218818544920457.jar tmpDir=null
17/12/18 20:55:48 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/18 20:55:49 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/18 20:55:49 INFO mapred.FileInputFormat: Total input paths to process : 1
17/12/18 20:55:49 INFO mapreduce.JobSubmitter: number of splits:2
17/12/18 20:55:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1513646413556_0007
17/12/18 20:55:50 INFO impl.YarnClientImpl: Submitted application application_1513646413556_0007
17/12/18 20:55:50 INFO mapreduce.Job: The url to track the job: http://hadoop:8088/proxy/application_1513646413556_0007/
17/12/18 20:55:50 INFO mapreduce.Job: Running job: job_1513646413556_0007
17/12/18 20:55:56 INFO mapreduce.Job: Job job_1513646413556_0007 running in uber mode : false
17/12/18 20:55:56 INFO mapreduce.Job:  map 0% reduce 0%
17/12/18 20:56:07 INFO mapreduce.Job:  map 21% reduce 0%
17/12/18 20:56:20 INFO mapreduce.Job:  map 60% reduce 0%
17/12/18 20:56:21 INFO mapreduce.Job:  map 100% reduce 100%
17/12/18 20:56:21 INFO mapreduce.Job: Job job_1513646413556_0007 completed successfully
17/12/18 20:56:22 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=223654
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=897876
                HDFS: Number of bytes written=16664198
                HDFS: Number of read operations=14
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=46214
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=46214
                Total vcore-milliseconds taken by all map tasks=46214
                Total megabyte-milliseconds taken by all map tasks=47323136
        Map-Reduce Framework
                Map input records=21
                Map output records=26
                Input split bytes=186
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=136
                CPU time spent (ms)=29950
                Physical memory (bytes) snapshot=511832064
                Virtual memory (bytes) snapshot=2122776576
                Total committed heap usage (bytes)=434110464
        File Input Format Counters
                Bytes Read=897690
        File Output Format Counters
                Bytes Written=16664198
17/12/18 20:56:22 INFO streaming.StreamJob: Output directory: /tmp/filee3c3c72924
17/12/18 20:56:25 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee3cede9c85
17/12/18 20:56:29 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee3c70739c88
> dataprocs <- from.dfs(proces)
> regpreproc <- dataprocs$val
> deppre <- table(presencialpreg$ESTADO_REGISTRO)
> deppre
```

Figura 31 – Resultado Modelo Estado Registro
Fuente: Autor
Elaboración: Autor

- Modelo para el objetivo 4.

```
> dat.values <- to.dfs(dat)
17/12/20 18:40:30 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
17/12/20 18:40:30 INFO compress.CodecPool: Got brand-new compressor [.deflate]
> proces <- mapreduce(input=dat.values)
17/12/20 18:41:06 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
packageJobJar: [/tmp/hadoop-unjar1304994391761479689/] [] /tmp/streamjob8861903643455487085.jar tmpDir=null
17/12/20 18:41:06 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/20 18:41:07 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/20 18:41:07 INFO mapred.FileInputFormat: Total input paths to process : 1
17/12/20 18:41:08 INFO mapreduce.JobSubmitter: number of splits:2
17/12/20 18:41:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1513807224474_0003
17/12/20 18:41:08 INFO impl.YarnClientImpl: Submitted application application_1513807224474_0003
17/12/20 18:41:08 INFO mapreduce.Job: The url to track the job: http://hadoop:8088/proxy/application_1513807224474_0003/
17/12/20 18:41:08 INFO mapreduce.Job: Running job: job_1513807224474_0003
17/12/20 18:41:14 INFO mapreduce.Job: Job job_1513807224474_0003 running in uber mode : false
17/12/20 18:41:14 INFO mapreduce.Job:  map 0% reduce 0%
17/12/20 18:41:25 INFO mapreduce.Job:  map 50% reduce 0%
17/12/20 18:41:33 INFO mapreduce.Job:  map 100% reduce 100%
17/12/20 18:41:33 INFO mapreduce.Job: Job job_1513807224474_0003 completed successfully
17/12/20 18:41:33 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=223656
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1918
                HDFS: Number of bytes written=1675
                HDFS: Number of read operations=14
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=34064
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=34064
                Total vcore-milliseconds taken by all map tasks=34064
                Total megabyte-milliseconds taken by all map tasks=34881536
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Input split bytes=186
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=89
                CPU time spent (ms)=13370
                Physical memory (bytes) snapshot=354807808
                Virtual memory (bytes) snapshot=2115088384
                Total committed heap usage (bytes)=370147328
        File Input Format Counters
                Bytes Read=1732
        File Output Format Counters
                Bytes Written=1675
17/12/20 18:41:33 INFO streaming.StreamJob: Output directory: /tmp/filee831d80f0a6
17/12/20 18:41:38 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee8330328199
17/12/20 18:41:42 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/filee8324b34980
> dataprocs <- from.dfs(proces)
> datos <- dataprocs$val
> datos
   Cod                      Matriculados
1    1                             11222
2    2                              1303
3    3                              3132
4    4                              1563
5    5                              2448
6    6                              3291
7    7                              8225
8    8                              2790
9    9                             12481
10  10                              3787
11  11                              9497
12  12                              1356
13  13                              5366
14  14                              2772
15  15                              1412
16  16                              1243
17  17                             53220
18  18                              3477
19  19                              1879
20  21                              1114
21  22                              2635
22  23                              2459
23  24                              5005
24  25                              1978
```

Figura 32 - Resultado Modelo Centros
Fuente: Autor
Elaboración: Autor

Descripción del modelo.

En esta sección se describe los resultados obtenidos de la ejecución de cada uno de los modelos de agrupamiento para cada objetivo, estos resultados se estudiarán más a fondo en la etapa de evaluación.

4.2.5. Evaluación (obtención de resultados).

Esta fase detalla la evaluación de los modelos generados, en esta ocasión se realiza una evaluación desde el punto de vista de negocio. Luego de realizar la evaluación se debe decidir si los objetivos se han cumplido y si esto se realizó satisfactoriamente se prosigue a la fase de implantación, caso contrario se debe identificar los errores encontrados y revisar nuevamente el proceso.

4.2.5.1. Evaluar los Resultados.

Para la ejecución de la evaluación de los resultados se realizó desde el punto de vista de negocio, como se estableció anteriormente en los objetivos la finalidad de este trabajo es el análisis y visualización de datos mediante la herramienta R juntamente integrado con Hadoop, para demostrar que los resultados son aceptables es necesario tener una base objetiva por cada uno de los interesados.

4.2.5.2. Revisar el Proceso.

El proceso de análisis y visualización de la data se ha ejecutado con los tiempos establecidos, han existido un poco de complicaciones hasta entender la estructura de los datos y como poder obtener los resultados. La causa de este retraso fue la familiarización con las herramientas R y Hadoop, ya una vez que se obtuvo el conocimiento adecuado y la forma de realización de las visualizaciones el trabajo no conllevo alguna complicación mayor.

4.2.5.3. Determinar los Próximos Pasos.

El siguiente paso que se desarrolla es la puesta en producción del prototipo que servirá para la presentación de los resultados a los diferentes interesados.

4.2.6. Despliegue (puesta en producción).

En esta fase de la metodología se procede a desarrollar un prototipo, el mismo que servirá para presentar el resultado final que es la generación de visualizaciones de la data a las autoridades competentes que necesiten analizar estos resultados, además en esta sección se explica cuál fue el proceso de desarrollo de la misma.

4.2.6.1. Planear el Despliegue.

Para la puesta en marcha de este proyecto primeramente fue necesario contar con la data a analizar, es decir la data que contiene la información necesaria para poder obtener los resultados y de estos obtener las visualizaciones requeridas. Luego se procede a la ejecución de los pasos de la metodología antes mencionados, partiendo desde la comprensión del negocio hasta el despliegue. Algunas fases como la comprensión y preparación de los datos fueron las más complejas y conllevó un tiempo adicional al planificado anteriormente. Al utilizar la herramienta R integrada con

Hadoop, fue necesario primeramente probar el correcto funcionamiento de las mismas, así mismo que los datos devueltos por Hadoop sean los correctos para su posterior análisis y visualización. Para cumplir con el último objetivo el cual es la generación de un mapa fue necesario buscar una librería de R que permita la generación de un mapa a partir de la lectura de unos archivos generados por el INEC y estos se puedan integrar con los datos extraídos del análisis del campo de centros educativos de la data obtenida por parte de la universidad. Una vez que se han cumplido con todos los objetivos del análisis y la visualización de la data, se procede al desarrollo de un prototipo que permita la presentación de los resultados a las diferentes autoridades que requieran de los mismos.

El prototipo es una aplicación desarrollada en PHP, la cual permite la ejecución de un script de R diseñado para cada una de las visualizaciones requeridas, dentro del script se realiza el análisis de los datos así mismo como el llamado a Hadoop y la generación de la visualización.

Arquitectura del Análisis de los Datos.

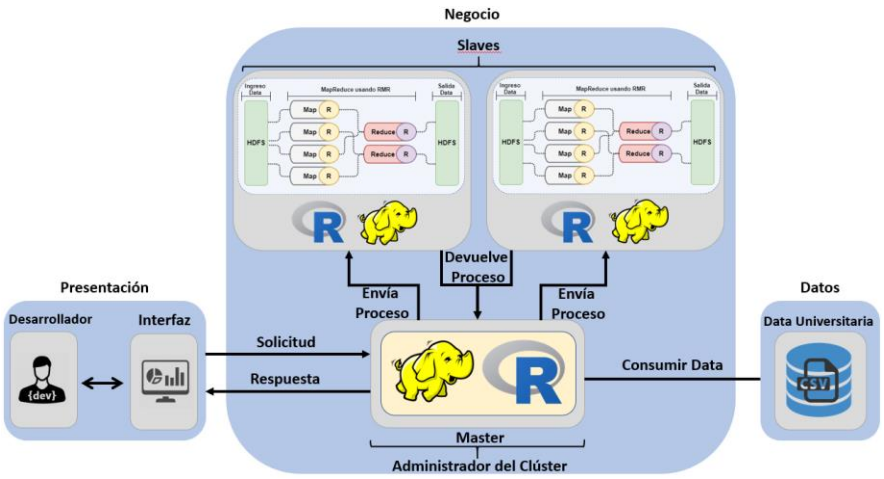


Figura 33 - Arquitectura Análisis de Datos
Fuente: Autor
Elaboración: Autor

En la capa de presentación el desarrollador se conecta a la interfaz de RStudio la cual se encuentra instalada en el computador master, dentro de este equipo se realizan las diferentes tareas de análisis y visualización.

En la capa de negocio se presenta la infraestructura montada y utilizada para el desarrollo de este trabajo, se representan 3 equipos con sistema operativo Ubuntu, dentro de cada uno se encuentra instalado Hadoop y R los mismos que se encuentran integrados, los equipos se dividen en un equipo que es el master y los restantes son los esclavos que son encargados de ayudar al procesamiento distribuido de la data, el

equipo master es el encargado de realizar todas las tareas de enviar el procesamiento a los esclavos, una vez que estos han realizado el procesamiento estos devuelven al master los resultados de dicho procesamiento, con el resultado de este procesamiento el master se encarga de realizar las diferentes operaciones de visualización de la data procesada.

La capa de datos representa los datos que se usan para el análisis procesamiento y visualización mediante la herramienta RHadoop, los datos son generados por los diferentes sistemas con los que cuenta la universidad, la data se encuentra almacenada en un archivo CSV el cual es llamado a la interfaz de RStudio y se trabaja directamente sobre ellos.

Desarrollo del análisis en RHadoop.

Como se mencionó en el capítulo anterior donde se detalla el planteamiento del problema, la data a analizar pertenece a información recolectada de uno de los sistemas que contiene la organización la cual se encuentra almacenada en un archivo CSV, cualquier persona interesada que desee observar los resultados se la va a ser muy complicado entenderla, es así como surge la necesidad de desarrollar este trabajo, el cual tiene el fin de analizar y visualizar los resultados mediante la utilización de las herramientas R y Hadoop la cuales se encuentran integradas para aprovechar las características de cada una.

Para comenzar con el análisis de los datos es necesario utilizar el programa RStudio, el cual gracias a su interfaz gráfica es de mucha ayuda para controlar y comprobar los resultados obtenidos paso a paso. El primer paso es ejecutar la aplicación RStudio.

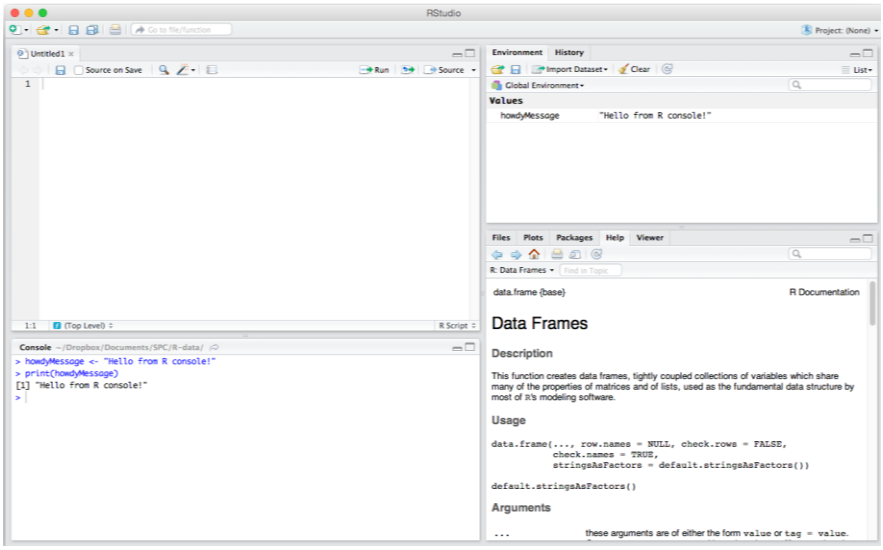


Figura 34 - Interfaz RStudio
Fuente: Autor
Elaboración: Autor

Cuando esté listo RStudio para la ejecución de un script se procede a realizar el llamado y ejecución del entorno de Hadoop, esto se realiza con los siguientes comandos, los cuales contienen la dirección donde se encuentra instalado y configurado Hadoop.

```
Sys.setenv(HADOOP_HOME="/usr/local/hadoop")
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar")
```

Figura 35 - Entorno de Hadoop Ejemplo1
Fuente: Autor
Elaboración: Autor

El siguiente paso es realizar el llamado a las librerías de Hadoop, las cuales sirven para la ejecución del proceso. Las librerías son llamadas de la siguiente manera:

```
# Librería necesaria para la ejecución de rhdfs y rmr2
library(rJava)
# Librería necesaria para la ejecución del HDFS de Hadoop
library(rhdfs)
# Librería necesaria para la ejecución del MapReduce de Hadoop
library(rmr2)
```

Figura 36 - Librerías de Hadoop Ejemplo1
Fuente: Autor
Elaboración: Autor

Lo siguiente que se realiza es cargar la data a analizar, es este caso la primera data a analizar se realiza mediante la ejecución de la siguiente línea:

```
datat <- read.csv(file="                                    2016.csv", head=TRUE, sep = ",")
```

Figura 37 - Lectura archivo CSV Ejemplo1
Fuente: Autor

Una vez que se tiene cargada la data a analizar y procesar se realiza la ejecución de los siguientes comandos los cuales sirven para realizar el procesamiento con Hadoop.

```
# Cargar los valores al HDFS de Hadoop
datat.values <- to.dfs(datat)
# Ejecutar el procesamiento de MapReduce de Hadoop
proces <- mapreduce(input=datat.values)
# Recuperar el procesamiento del MapReduce de Hadoop
datproc <- from.dfs(proces)
# Recuperar los datos procesados por Hadoop
dataprc <- datproc$val
```

Figura 38 - Procesamiento Hadoop Ejemplo1
Fuente: Autor
Elaboración: Autor

Cuando se haya recuperado la data procesada se procede a separarla por atributos internos con los siguientes comandos:

Figura 39 - Filtrar Data Ejemplo1
Fuente: Autor
Elaboración: Autor

Para comprobar que los datos han sido separados correctamente se realiza la ejecución de los siguientes comandos:

```
# Comprobación de los datos correctos Modalidad Presencial
view(pres)
# Comprobación de los datos correctos Modalidad Distancia
view(dist)
```

Figura 40 - Visualización Resultado Ejemplo1
Fuente: Autor
Elaboración: Autor

Una vez que se haya comprobado que los datos se han separado correctamente es necesario nuevamente realizar un procesamiento con Hadoop para cada una de las modalidades esto se realiza con los siguientes comandos:

```
# Cargar los valores al HDFS de Hadoop
pres.values <- to.dfs(pres)
# Ejecutar el procesamiento de MapReduce de Hadoop
proces <- mapreduce(input=pres.values)
# Recuperar el procesamiento del MapReduce de Hadoop
datproc <- from.dfs(proces)
# Recuperar los datos procesados por Hadoop
dataprc <- datproc$val
```

Figura 41 - Proceso 2 Hadoop Ejemplo1
Fuente: Autor
Elaboración: Autor

Una vez procesada cada una de sus atributos generales se procede a separar la data analizada y procesada por otros atributos dependientes de los anteriores, en este caso se realiza de la siguiente manera:

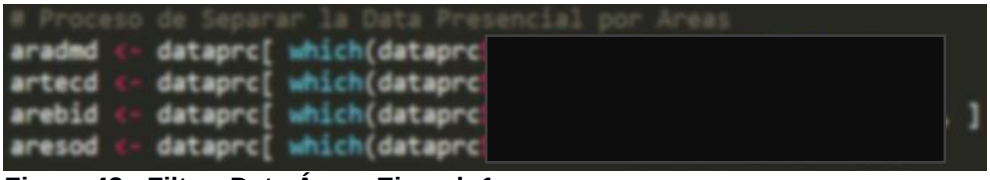


Figura 42 - Filtrar Data Áreas Ejemplo1
Fuente: Autor
Elaboración: Autor

Luego de realizar la separación de la data por otros atributos se procede a analizar qué resultados se puede obtener de esta separación. A continuación, se detalla cómo se realizó la generación de una gráfica. La gráfica por demostrar es una de resultados generales. Como primer paso se procesa la data con MapReduce de Hadoop mediante las siguientes líneas de código:

```
# Cargar los valores al HDFS de Hadoop
aradmd.values <- to.dfs(aradmd)
# Ejecutar el procesamiento de MapReduce de Hadoop
proces <- mapreduce(input=aradmd.values)
# Recuperar el procesamiento del MapReduce de Hadoop
datproc <- from.dfs(proces)
# Recuperar los datos procesados por Hadoop
aradprc <- datproc$val
```

Figura 43 - Proceso 3 Hadoop Ejemplo1
Fuente: Autor
Elaboración: Autor

Recuperado el procesamiento de los datos por Hadoop se realiza en análisis de los mismos y se genera una gráfica con los siguientes comandos:

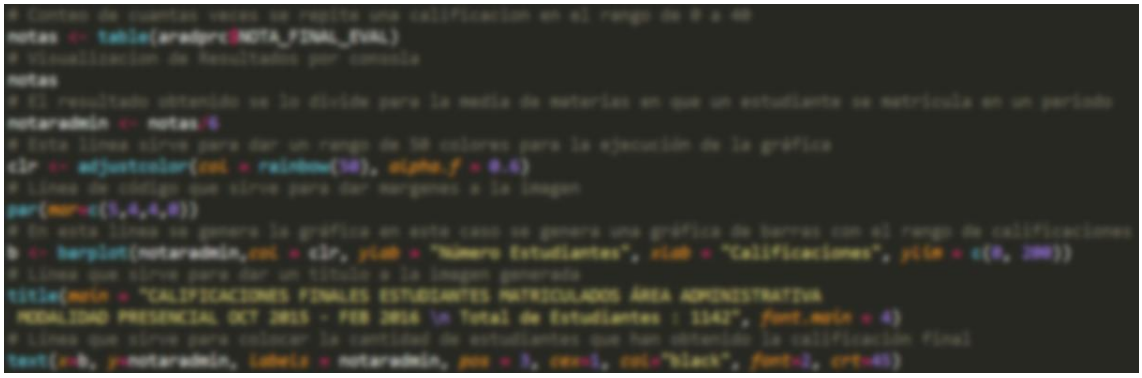


Figura 44 - Proceso Gráfica Ejemplo1
Fuente: Autor
Elaboración: Autor

Una vez realizado los pasos anteriores se obtiene la siguiente imagen:

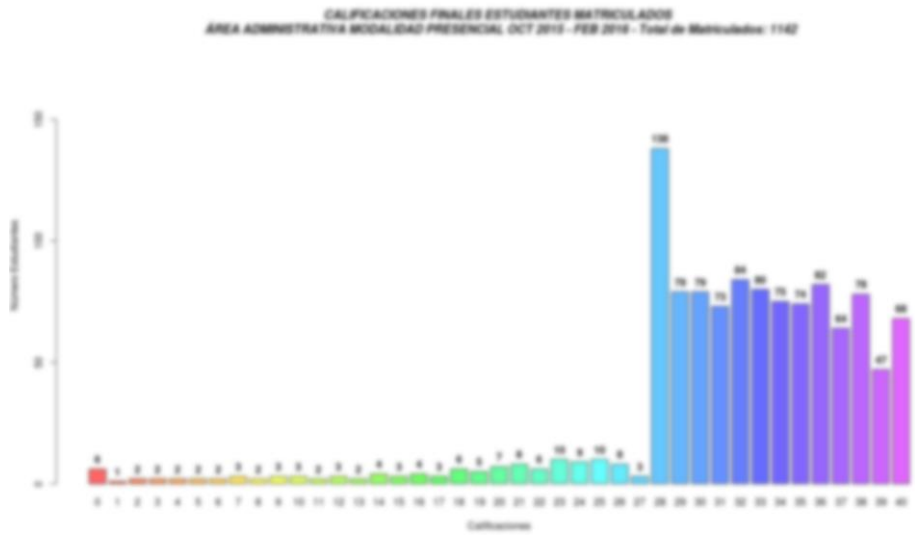


Figura 45 - Gráfica Ejemplo1
Fuente: Autor
Elaboración: Autor

Para el análisis y posterior creación de la visualización del mapa se realiza el siguiente proceso, se inicia llamando al entorno de Hadoop de la siguiente manera:

```
Sys.setenv(HADOOP_HOME="/usr/local/hadoop")
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar")
```

Figura 46 - Llamar Entorno Hadoop Ejemplo2
Fuente: Autor
Elaboración: Autor

Se procede a llamar a las librerías de Hadoop de la siguiente manera:

```
# Librería necesaria para la ejecución de rhdfs y rmr2
library(rJava)
# Librería necesaria para la ejecución del HDFS de Hadoop
library(rhdfs)
# Librería necesaria para la ejecución del MapReduce de Hadoop
library(rmr2)
```

Figura 47 - Librerias Hadoop Ejemplo2
Fuente: Autor
Elaboración: Autor

Luego se realiza el llamado de las librerías que permiten la lectura de los archivos generados por el INEC para la realización del mapa en R, estas librerías se llaman de la siguiente manera:

```
# Librerìas:
library(rgdal)
library(RColorBrewer)
library(classInt)
```

Figura 48 - Librerías Utilizadas Ejemplo2
Fuente: Autor
Elaboración: Autor

Se procede a cargar los archivos que servirán para la generación del mapa de la siguiente manera:

```
# Cargar archivos necesarios para realizacion de mapa
poligonos <- readOGR("/opt/lampp/htdocs/tesis/R/2012_nacional+por+provincias/nxprovincias.shp",layer="nxprovincias")
poligonos = poligonos[poligonos$DPA_PROVIN !=90, ]
centroides <- coordinates(poligonos)
```

Figura 49 - Archivos Necesarios Mapa Ejemplo2
Fuente: Autor
Elaboración: Autor

Se carga la data principal de la siguiente manera:

```
datat <- read.csv(file="[redacted]", head=TRUE, sep = ",")
```

Figura 50 - Lectura Archivo CSV Ejemplo2
Fuente: Autor
Elaboración: Autor

Se procede a realizar el procesamiento de la data con Hadoop y se recupera su resultado de la siguiente manera:

```
# Cargar los valores al HDFS de Hadoop
datat.values <- to.dfs(datat)
# Ejecutar el procesamiento de MapReduce de Hadoop
proces <- mapreduce(input=datat.values)
# Recuperar el procesamiento del MapReduce de Hadoop
datproc <- from.dfs(proces)
# Recuperar los datos procesados por Hadoop
dataprc <- datproc$val
```

Figura 51 - Procesamiento con Hadoop Ejemplo2
Fuente: Autor
Elaboración: Autor

Cuando se haya recuperado el resultado del procesamiento se realiza el filtrado de los datos por centro de la siguiente manera:

```
# Filtrado de datos por centro universitario
matprov <- table(
```

Figura 52 - Filtar Datos por Centro Ejemplo2
Fuente: Autor
Elaboración: Autor

Para visualizar el resultado obtenido realizamos la ejecución del siguiente comando:

```
# Verificar los datos de los centros
matprov
```

Figura 53 - Verificar Datos Ejemplo2
Fuente: Autor
Elaboración: Autor

Se obtienen los siguientes resultados a nivel general:

CENTRO	CENTRO	CENTRO
810	362	2651
308	5840	4640
3477	562	7458
1615	1143	2983
558	543	185
431	3821	3
214	750	430
938	208	7
1106	39	1848
380	52	1321
1213	138	205
410	12	479
43	52	96
1892	17	462
9288	115	542
591	1758	328
369	102	727
852	1076	4588
352	621	130
1972	2263	389
106	706	124
1235	470	372
756	575	409
1089	330	1060
6809	901	1563
1891	222	162
163	2129	329
624	1243	1229
3081	1194	1029
567	339	215
417	21466	
39	9068	
2448	6042	
579		

Figura 54 - Muestra Datos Ejemplo2
Fuente: Autor
Elaboración: Autor

Con el resultado antes obtenido se realiza la clasificación de tal forma que quede de la siguiente manera:

	Cod	Nombre.de.la.Provincia	Matriculados
1	1		11222
2	2		1303
3	3		3132
4	4		1563
5	5		2448
6	6		3291
7	7		8225
8	8		2790
9	9		12481
10	10		3787
11	11		9497
12	12		1356
13	13		5366
14	14		2772
15	15		1412
16	16		1243
17	17		53220
18	18		3477
19	19		1879
20	21		1114
21	22		2635
22	23		2459
23	24		5005
24	25		1978

Figura 55- Resultado Centros por Provincia Ejemplo2
Fuente: Autor
Elaboración: Autor

Luego de haber organizado y los resultados por provincia se realiza el proceso de enlazar los datos con los archivos descargados del INEC y descritos en pasos anteriores, igualmente se enlaza con los acrónimos generados, posteriormente los resultados son divididos para la media, esto se realiza de la siguiente manera:

```
# Enlazar los datos de matriculados por provincia con los datos del mapa
datos$Cod <- poligonos@data
# Enlazar de los valores de matriculados con los acrónimos
LlarC <- as.matrix(datos[,3])
rownames(LlarC) <- AcrProv
# Extrancción de los datos de matriculados
matri <- datos[,3]
# División de los datos analizados de estudiantes matriculados por provincia
# para media de componentes seleccionados por los estudiantes matriculados en el periodo
matri <- matri/6
matri <- round(matri)
```

Figura 56- Enlazadar Datos Ejemplo 2
Fuente: Autor
Elaboración: Autor

Luego de realizar el paso anterior se procede nuevamente a enlazar los datos para posteriormente realizar la gráfica.

```r
# Enlazar los datos para realizar la gráfica
matri <- as.data.frame(matri)
names(matri) <- "matri"
row.names(matri) <- row.names(poligonos)
poligonos.data <- SpatialPolygonsDataFrame(poligonos,matri)
plotvar <- poligonos.data$matri
nclr <- 8 # Numero de colores
plotclr <- brewer.pal(nclr,"Blues")
class <- classIntervals(round(plotvar,1),nclr,style="quantile")
colcode <- findColours(class,plotclr) # define paleta de colores
```

Figura 57 - Operaciones Datos Enlazados Ejemplo2
Fuente: Autor
Elaboración: Autor

Una vez finalizado la ejecución de los pasos anteriores se procede a generar la gráfica del mapa de la siguiente manera:

```r
# Generar gráfica del mapa
plot (poligonos.data, col=colcode, border="grey", axes=T, xaxt="n", yaxt="n")
# Asignar un titulo a la gráfica
title(main = "Proporción de Estudiantes Matriculados Modalidad a Distancia por Provincia - Oct 2015 - Feb 2016",cex=3)
# Asignar leyenda del rango de estudiantes
legend(title = "Rango Estudiantes","bottomleft", c("Desde 186 hasta 225","Desde 226 hasta 300", "Desde 301 hasta 409","Desde 410 hasta 464",
  "Desde 465 hasta 560", "Desde 561 hasta 849", "Desde 850 hasta 1619", "Desde 1620 hasta 8870"),  fill= attr(colcode,"palette"),cex=0.8)
# Asignar acrónimos a las provincias
text(centroides,AcrProv,cex=1)
```

Figura 58 - Generación Mapa Matriculados Ejemplo2
Fuente: Autor
Elaboración: Autor

Y como resultado se obtiene la siguiente imagen del mapa:

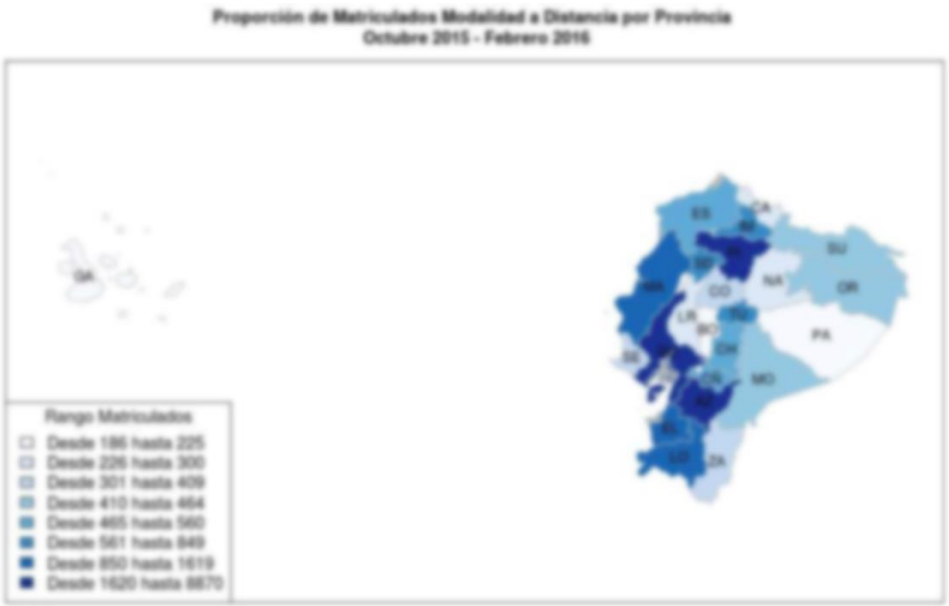


Figura 59 - Gráfica Ejemplo2
Fuente: Autor
Elaboración: Autor

Desarrollo del Prototipo para mostrar los resultados del análisis en RHadoop.

Las herramientas que se utilizaron para el desarrollo del prototipo son:

- XAMMP.
- PHP.
- R.
- Sublime.
- Hadoop.

Al desarrollar el prototipo en PHP se tiene la ventaja de que permite la ejecución de script diseñado para cada una de las visualizaciones además de que permite la ejecución en segundo plano de Hadoop. Esto significa que el desarrollador se encarga de que el funcionamiento del prototipo sea el correcto, ya que anteriormente se ha probado cada uno de los scripts para la realización de cada análisis y visualización de los datos.

Arquitectura del Prototipo.

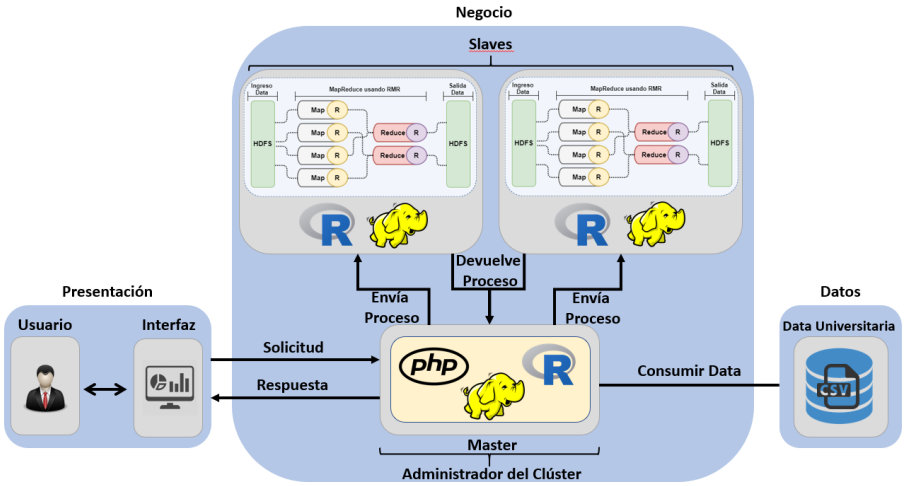


Figura 60 - Arquitectura del Prototipo
Fuente: Autor
Elaboración: Autor

En la capa de presentación se representa al usuario que utiliza el prototipo, el ingreso al prototipo se realiza mediante un navegador Web con la dirección **localhost/tesis/index.php**, el navegador realiza un llamado al servidor y el mismo le envía una respuesta al usuario presentando la interfaz del prototipo en el navegador.

Cuando el usuario ingrese a una de las tres opciones presentadas para la visualización de resultados, se le presenta un formulario el cual debe llenar detenidamente para seleccionar el tipo de resultado que desea obtener, una vez que se ha llenado el formulario el usuario envía la petición de generación de la gráfica e internamente en segundo plano se realiza la ejecución de un script desarrollado en R para el tipo de gráfica que el usuario solicita, este proceso se realiza en la capa de negocio en la cual se detalla la infraestructura desarrollada para el análisis y

visualización de la data la cual se detalló anteriormente, una vez ya realizado el procesamiento de la solicitud de obtención de la gráfica, se ejecuta un script en el que se realiza la lectura del archivo CSV de la capa de datos, internamente trabaja R y genera una gráfica la cual se almacena en una dirección dada, el prototipo hace lectura de la imagen generada y es presentada al usuario que la solicitó.

Diseño y Desarrollo del Prototipo.

Para la presentación de los resultados del análisis y procesamiento de la data se realiza un prototipo web, el que permitirá al usuario visualizar un resultado que el desee, esto se hace posible a partir de la ejecución en segundo plano del script que contiene el análisis y visualización de la data en el lenguaje R integrado juntamente con Hadoop.

El diseño y desarrollo del prototipo debe quedaría de la siguiente manera:

- **Pantalla principal** con tres botones principales que dirigen a otra página, dichos botones permiten al usuario poder escoger que tipo de resultados desea visualizar, los cuales son:
    - Resultados Generales.
    - Resultados por Titulaciones.
    - Resultados por Mapa.

  Siguiendo lo mencionado anteriormente la pantalla principal del prototipo queda de la siguiente manera:

Figura 61 - Interfaz Principal del Prototipo
Fuente: Autor
Elaboración: Autor

El código fuente del desarrollo de la página anterior es la siguiente:

```php
<!DOCTYPE html>
<html>
    <head>
        <title>RHadoop</title>
        <LINK REL=StyleSheet HREF="css/css.css" TITLE="Loading">
    </head>

    <body style="background-color:#F2F2F2">

        <div class="header1">
            <img id="cabecera" src="images/utpl2.jpg">
        </div>
        <h1 align="center">Prototipo de Análisis y Visualización con Rhadoop</h1>
        <h2 align="center">Sección Departamental: Tecnologías Avanzadas de la Web y Sistemas Basados en Conocimiento</h2
        >
        <h2 align="center">Titulación: Sistemas Informáticos y Computación</h2>
        <h2 align="center">Desarrollado por: Santiago Merino</h2>

        <section id="menu">
            <div class="general">
                <img id="imgg" src="images/general.png">
                <a href="generales.php"><button id="btn1" type="button">General</button></a>
            </div>
            <div class="titulacion">
                <img id="imgg" src="images/titulacion.png">
                <a href="titulaciones.php"><button id="btn2" type="button">Titulación</button></a>
            </div>
            <div class="mapa">
                <img id="imgg" src="images/mundo.png">
                <a href="mapa.php"><button id="btn3" type="button">Mapa</button></a>
            </div>

        </section>

    </body>
</html>
```

Figura 62 - Código de la Página Principal
Fuente: Autor
Elaboración: Autor

- **Página resultados generales,** en esta página el usuario se encuentra con un formulario en el cual debe escoger cada una de las opciones para poder visualizar el resultado general deseado, el diseño de la página de resultados generales queda de la siguiente manera:

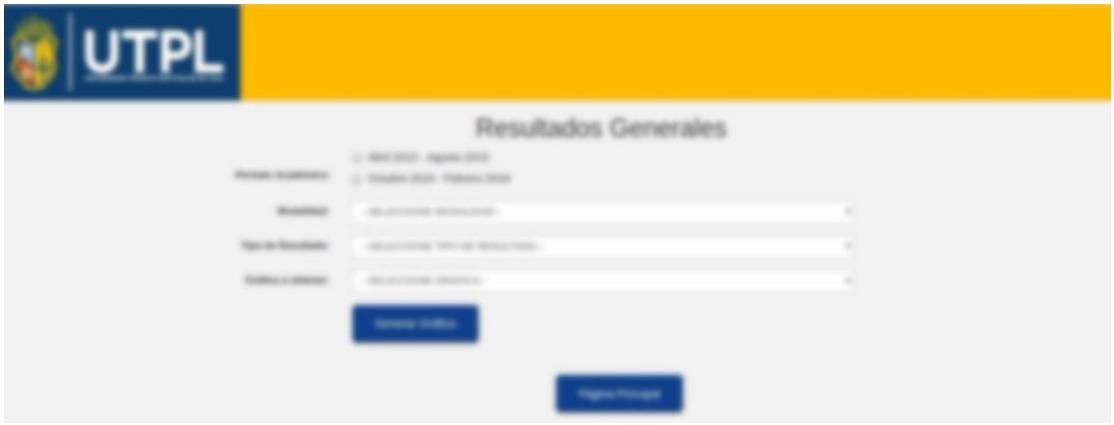


Figura 63 - Interfaz Resultados Generales
Fuente: Autor
Elaboración: Autor

Además, se cuenta con un botón que permite al usuario volver a la página principal que le permita escoger otro tipo de resultado que desee visualizar.

- **Página resultados por titulaciones,** igualmente descrito anteriormente el usuario en esta página se encuentra con un formulario que debe llenar con las

opciones deseadas por él mismo y así obtener la visualización deseada, el diseño de la página de resultados por titulación queda de la siguiente manera:

Figura 64 - Interfaz Resultados por Titulación
Fuente: Autor
Elaboración: Autor

Además, se cuenta con un botón que permite al usuario volver a la página principal que le permita escoger otro tipo de resultado que desee visualizar.

- **Página resultados por mapa,** en esta página el usuario se encuentra con un formulario de una opción en el cual debe seleccionar el periodo académico para generar un mapa de los estudiantes matriculados en la modalidad a distancia a nivel nacional por provincia, el diseño de la página del resultado de matriculados a distancia queda de la siguiente manera:

Figura 65- Interfaz Mapa Matrículados a Distancia
Fuente: Autor
Elaboración: Autor

Además, se cuenta con un botón que permite al usuario volver a la página principal que le permita escoger otro tipo de resultado que desee visualizar.

4.2.6.2. Planear la Monitorización y Mantenimiento.

La supervisión y mantenimiento de la implementación que se ha desarrollado del presente trabajo es una de las fases más importantes, debido a que los datos generados por los diferentes sistemas con los que cuenta la universidad pueden ser

modificados por alguna razón ya sea una confusión o solución de calificaciones de los estudiantes, es así que se debe estar al tanto de estos cambios para q la data a trabajar siempre este actualizada y los resultados que se obtienen sean lo más actual posible.

El plan de supervisión y mantenimiento que se podría dar a este trabajo se detalla:

- Supervisar semestralmente que los datos con los que se trabaja sean lo más actual posibles a fin de evitar generación de resultados obsoletos o equivocados.
- Supervisar periódicamente que la infraestructura diseñada en este trabajo se encuentre funcionando correctamente y su disponibilidad sea ininterrumpida.
- Supervisar periódicamente que el prototipo esté funcionando correctamente y se encuentre disponible en todo momento.

CAPITULO V PRUEBAS DE VALIDACIÓN

5.1. Introducción.

La realización de un plan de pruebas es una de las tareas más importantes en el desarrollo de un proyecto, ya sea de análisis de datos o de software, al realizar una planificación de pruebas se asegura que el trabajo que se está realizando es un trabajo de calidad.

En este capítulo se busca describir los puntos más relevantes del plan de pruebas de este trabajo de titulación, a continuación, se describe cada una de las pruebas ejecutadas y de los resultados obtenidos.

5.2. Ambiente de Pruebas.

Para la ejecución de las pruebas unitarias a realizarse los equipos deben estar completamente listos y funcionales, así se obtendrán los resultados reales de cada una de las pruebas a realizarse.

Cabe recalcar que las pruebas se ejecutan en tres arquitecturas diferentes, dos arquitecturas pertenecen a máquinas virtuales, una multinodo y otro nodo singular, frente a una arquitectura nodo singular configurada en una máquina nativa, en la arquitectura multinodo de Hadoop, un equipo es el master y los dos equipos restantes son los nodos que se encargan de realizar el procesamiento. Las características de los equipos y programas utilizados son las siguientes:

Tabla 11 - Ambiente en las que se realizaron las Pruebas

Características	Multinodo Máquinas Virtuales			Nodo Singular Máquina Virtual	Nodo Singular Máquina Nativa
	Equipo Master	Equipo Slave1	Equipo Slave2	Equipo Slave1	Equipo Slave1
Sistema Operativo	Ubuntu 14.04	Ubuntu 14.04	Ubuntu 14.04	Ubuntu 14.04	Ubuntu 14.04
Versión Hadoop	2.6.5	2.6.5	2.6.5	2.6.5	2.6.5
Versión R	3.4.2	3.4.2	3.4.2	3.4.2	3.4.2
Versión RStudio	1.1.383	1.1.383	1.1.383	1.1.383	1.1.383
Versión HDFS	1.0.8	1.0.8	1.0.8	1.0.8	1.0.8
Versión MapReduce	3.3.1	3.3.1	3.3.1	3.3.1	3.3.1

Fuente y Elaboración Propia.

5.3. Ejecución de Pruebas.

Una vez que se tenga preparada la arquitectura y el ambiente a robar se procede a realizar cada una de las pruebas planteadas, cuyo objetivo principal es descubrir

errores que deben ser corregidos para asegurar el correcto funcionamiento de la arquitectura y calidad del sistema multinodo, para conocer más a detalle el resultado de cada una de las pruebas revisar el anexo 7.

5.3.1. Pruebas Unitarias.

Las pruebas unitarias permiten comprobar la correcta funcionalidad del sistema y la arquitectura multinodo, estas pruebas consisten en ejecutar una serie de peticiones al sistema y se debe obtener una respuesta favorable comprobando con el correcto funcionamiento de la arquitectura.

Tabla 12 - Ejecución Pruebas Unitarias

Requerimiento	Entrada	Salida	Tiempo de Ejecución	Resultado
Comunicación entre equipos de la arquitectura multinodo	Comando de comunicación ping "dirección Ip"	Respuesta de dirección Ip	5 segundos	Exitoso
Levantar Hadoop multinodo	• Comando start-dfs.sh • Comando start-yarn.sh	Respuesta satisfactoria	1 minuto	Exitoso
Comprobar en consola servicios Hadoop multinodo levantados	Ejecutar comando jps en terminal de los equipos.	• En master: ✓ SecondaryNamenode ✓ ResourceManager ✓ Namenode • En slaves: ✓ NodeManager ✓ DataNode	5 segundo	Exitoso
Comprobar interfaz de Hadoop	Ingresar la dirección: • master:50070 • master:8088	Interfaz de información de Hadoop y aplicaciones MapReduce en navegador	5 segundos	Exitoso
Ejecutar ejemplo funcionamiento de Hadoop multinodo	Ejecución ejemplo en terminal	Resultado de ejecución del ejemplo	1 minuto	Exitoso
Comprobar ejecución ejemplo en interfaz de aplicaciones Hadoop	Ingresar a la dirección master:8088	Registro del ejemplo ejecutado	1 minuto	Exitoso
Comprobar funcionamiento de R	Ingreso a R desde consola y ejecutar ejemplo	Consola de R y resultado del ejemplo	15 segundos	Exitoso

Comprobar funcionamiento de RStudio	Ingreso a la interfaz de R y ejecutar ejemplo	Interfaz de RStudio y resultado del ejemplo	15 segundos	Exitoso
Levantar entorno de Hadoop en RStudio	Ingresar comandos del entorno de Hadoop	Respuesta en consola de RStudio	10 segundos	Exitoso
Comprobar funcionamiento de las librerías de RHadoop en RStudio	Ingreso de librerías rhdfs y rmr2	Respuesta satisfactoria en consola de RStudio	10 segundos	Exitoso
Comprobar que la Data a trabajar se cargue correctamente en RStudio	Comando de lectura de Data	Visualización de la Data a Trabajar	20 segundos	Exitoso
Comprobar procesamiento de la Data con librerías de RHadoop en RStudio	Comandos de procesamiento de la data con RHadoop	Visualización del resultado del procesamiento en consola de Rstudio	1 minuto	Exitoso
Recuperar el procesamiento de la data y generar una gráfica	Comandos de recuperación y generación de gráfica	Gráfica generada	20 segundos	Exitoso

Fuente y Elaboración Propia.

5.3.2. Pruebas de Sistema.

Las pruebas de sistema permiten comprobar que la arquitectura de Hadoop funcione de manera correcta, y que el resultado del procesamiento de la herramienta RHadoop devuelva los resultados esperados y generen las gráficas correctamente. El ambiente de las pruebas de sistema se ha realizado en arquitecturas con máquinas virtuales y una con sistema operativo nativo con el sistema operativo Ubuntu 14.04, las pruebas se realizan con la arquitectura Hadoop nodo singular versus Hadoop multinodo, permitiendo comprobar el tiempo de respuesta de cada una de las arquitecturas. En resumen, las pruebas de sistema se presentan en la siguiente gráfica.

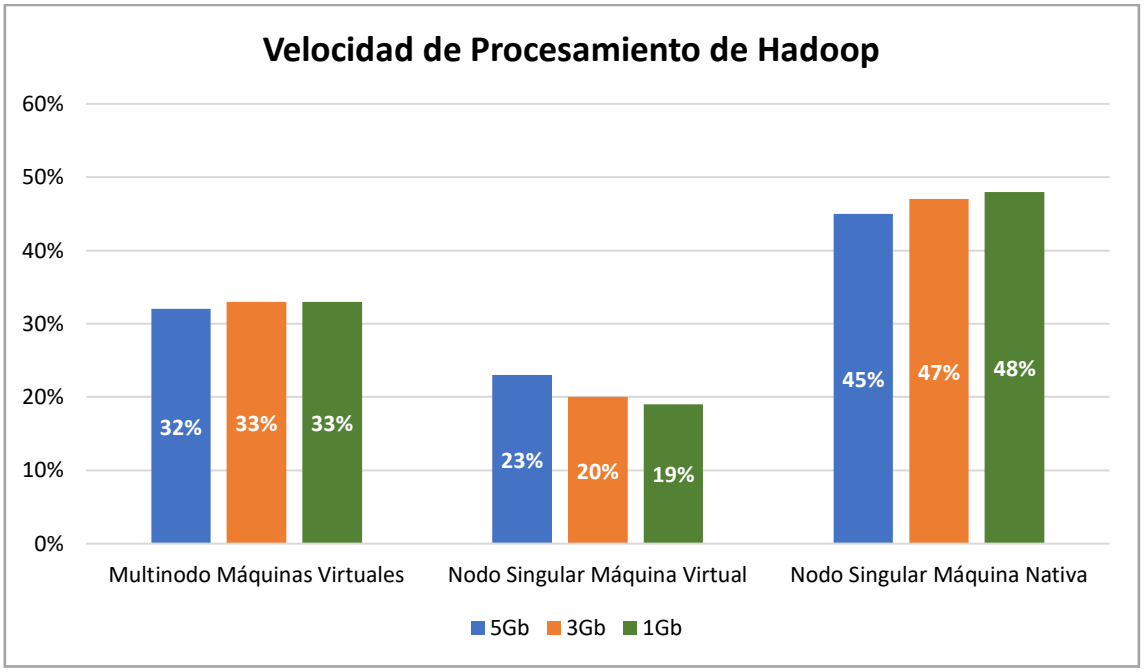


Figura 66 - Comparación Velocidad Hadoop
Fuente: Autor
Elaboración: Autor

Los resultados presentados en la figura 66 indican la comparación de velocidad del procesamiento por parte de MapReduce a un tamaño de la data procesada, se pueden interpretar que la arquitectura más veloz es una arquitectura de nodo singular en una máquina nativa, debido a que el equipo tiene mejores características y esto permite un mayor poder de procesamiento y velocidad que una máquina virtual obteniendo un promedio de 10% a 15% más veloz que una arquitectura multinodo en máquinas virtuales. Pero en comparación a la prueba en equipos virtuales se demuestra que el procesamiento es mucho más veloz en una arquitectura multinodo en comparación a una de nodo singular obteniendo una diferencia de 10% a 14% en velocidad de procesamiento. Para una mejor comprensión de los resultados de la figura 73 se puede revisar el anexo 7 apartado pruebas de sistema.

5.3.3. Pruebas de Caja Negra.

Las pruebas de caja negra permiten verificar la funcionalidad de cada uno de los scripts sin tomar en cuenta su estructura interna de código. Estas pruebas tienen la finalidad de enfocarse solamente en los datos de ingreso y las salidas esperadas por parte del interesado.

Para la ejecución de las pruebas de caja negra se realizó la ejecución de 50 pruebas entre las cuales se escogió una para presentar en este documento, para detallar un ejemplo de caja negra se tiene el siguiente requerimiento escogido al azar.

El resultado de un requerimiento es el siguiente:

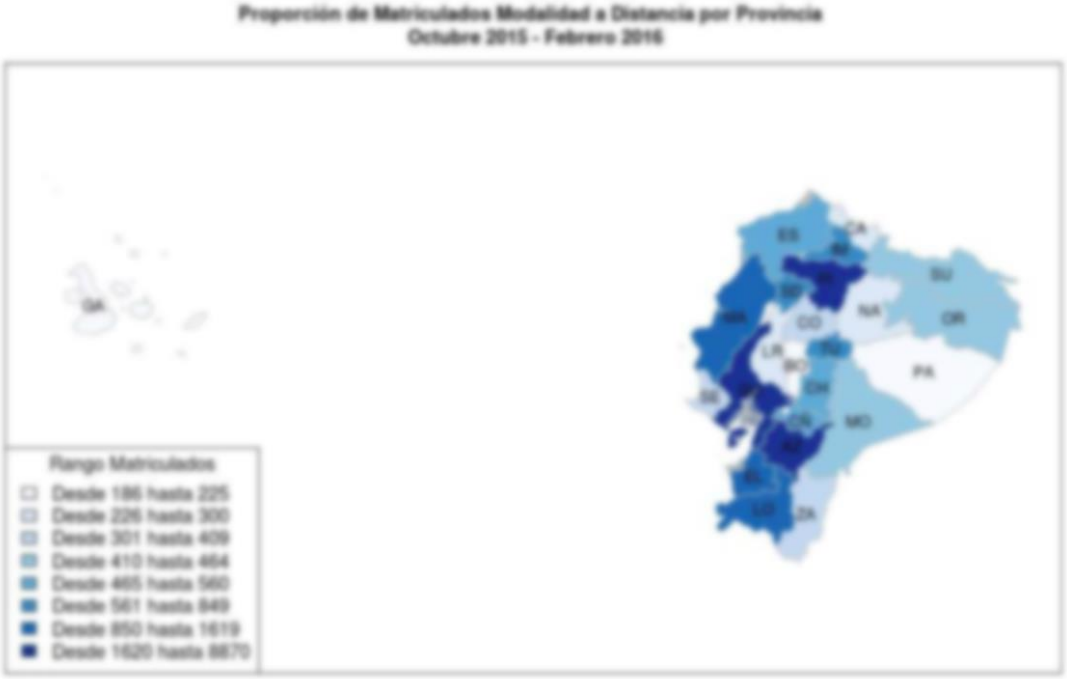


Figura 67 - Resultado Requerimiento
Fuente: Autor
Elaboración: Autor

Con la presentación al usuario final de la figura 67, se determina que el requerimiento solicitado se cumplió satisfactoriamente.

5.3.4. Pruebas de Rendimiento.

El objetivo de este tipo de pruebas es comprobar el tiempo de respuesta que se tiene de la arquitectura de Hadoop, para cumplir con esta prueba se toma los tiempos en que tarda un script en devolver el resultado solicitado, el script se encuentra diseñado en R que contiene la integración de RHadoop.

Para cumplir con este objetivo se realiza la ejecución de un script en donde se compara el tiempo de respuesta de una arquitectura de Hadoop multinodo versus Hadoop nodo singular, los resultados a evaluar son los siguientes:

- Generar una gráfica de resultados generales.
- Generar una gráfica de resultados por titulación.
- Generar una gráfica de un mapa.

Se realiza un promedio de 20 consultas por cada uno de los resultados anteriormente descritos, y el tiempo promedio de respuesta es:

Tabla 13 - Resultados por Tiempo

Tipo de Resultado	Hadoop Multinodo Máquinas Virtuales	Hadoop Nodo Singular Máquinas Virtual	Hadoop Nodo Singular Máquina Nativa
Generar una gráfica de resultados generales	1 minuto 42 segundos	2 minutos 20 segundos	1 minuto 19 segundos
Generar una gráfica de resultados por titulación	1 minuto 40 segundos	2 minutos 12 segundos	58 segundos
Generar una gráfica de un mapa	1 minuto 18 segundos	1 minuto 56 segundos	1 minuto 2 segundos

Fuente y Elaboración Propia

Se debe considerar que los resultados anteriormente presentados pueden variar, ya que se realizó varias pruebas y el tiempo presentado es un tiempo promedio y puede variar de acuerdo con el tamaño de la data con la que se trabaja.

5.3.5. Prueba de interfaz de usuario.

Las pruebas de interfaz de usuario determinan que el prototipo desarrollado ofrece a los usuarios finales una interfaz amigable en la que puedan interactuar fácilmente y obtener los resultados esperados. Las pruebas se realizan en la herramienta diseñada por W3C, la cual ofrece dos funcionalidades de validación, una para la estructura HTML y la segunda para CSS, de la siguiente manera:

Tabla 14 - Herramientas de Validación

Herramienta	Acción	Resultado
W3C Markup Validator Service	Validar la estructura HTML del prototipo	Exitoso
W3C CSS Validator Service	Validar el CSS	Exitoso

Fuente y Elaboración Propia

El resultado de las pruebas de validación de HTML en W3C es la siguiente:

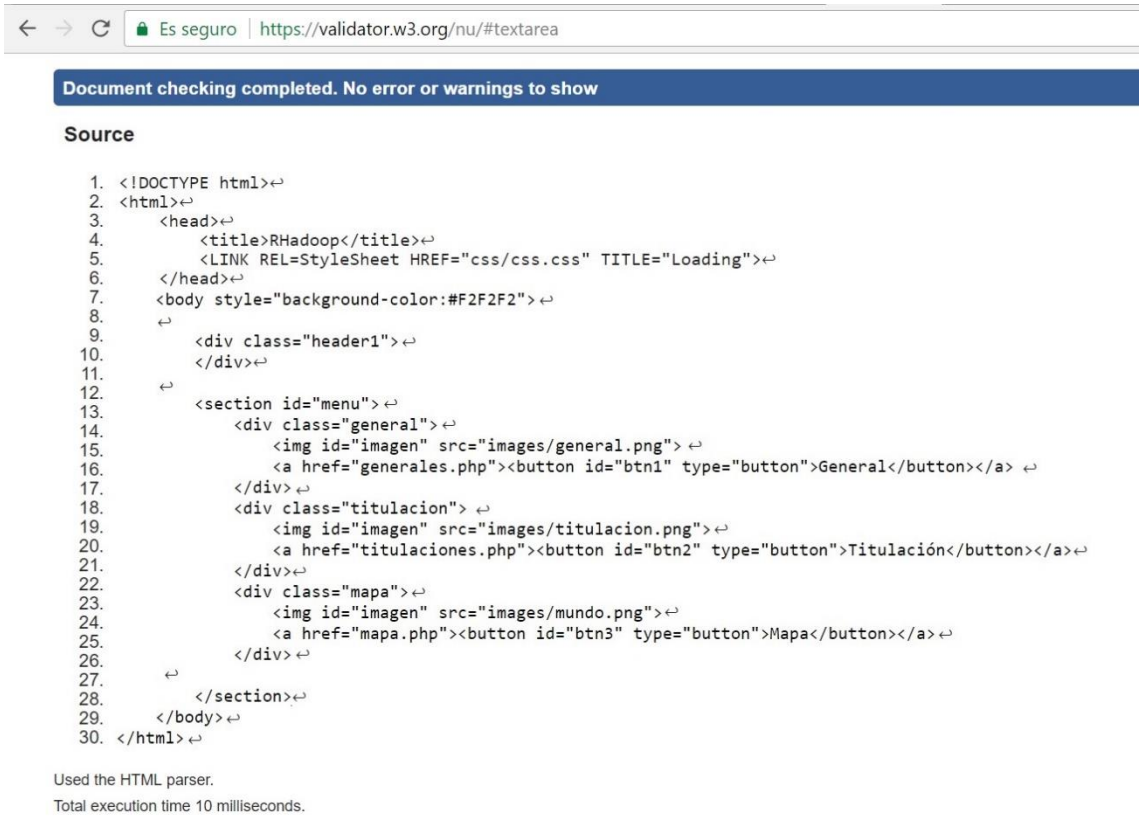


Figura 68 - Validación HTML
Fuente: Autor
Elaboración: Autor

El resultado de las pruebas de validación del CSS en W3C es la siguiente:

Figura 69 - Validación CSS
Fuente: Autor
Elaboración: Autor

5.3.6. Pruebas de Calidad del Software.

La calidad del software involucra una gran cantidad de aspectos o características que determinan la utilidad y capacidad de satisfacer la necesidad de los clientes con eficiencia, flexibilidad, confiabilidad, portabilidad, mantenibilidad, usabilidad, integridad y seguridad.

Para realizar las pruebas del código PHP desarrollado para cumplir con el correcto funcionamiento del prototipo se realizó en SonarQube, herramienta que consiste en evaluar el código fuente y emitir resultados para generar un reporte de calidad del software, obteniendo los siguientes resultados:

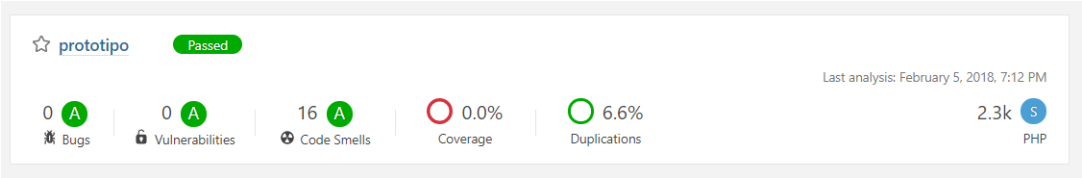


Figura 70 - Resultado General SonarQube
Fuente: Autor
Elaboración: Autor

Con los resultados obtenidos se determina que el prototipo cumple con las principales características de un software como lo es la Confiabilidad, Seguridad y Mantenibilidad, ya que aprueba exitosamente cada una de las pruebas realizadas por SonarQube, para más detalle se presenta la siguiente imagen:

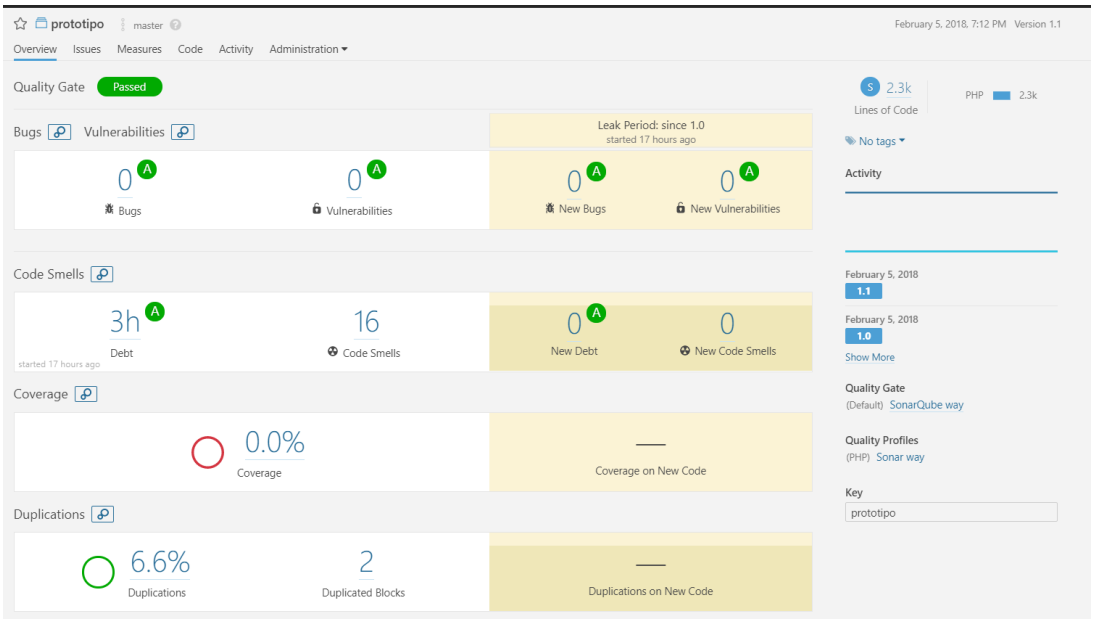


Figura 71 - Resultado Detallado SonarQube
Fuente: Autor
Elaboración: Autor

5.4. Análisis de resultados.

Con los resultados que se obtienen de cada una de las pruebas se puede determinar que:

- Se garantiza que la construcción y configuración de la arquitectura multinodo de Hadoop es el correcto y que los resultados que esta ofrece son óptimos.
- Se garantiza que con la ejecución de cada una de las pruebas las herramientas que se encuentran instaladas en la arquitectura multinodo de Hadoop funcionan correctamente y se integran correctamente.
- Se garantiza que la arquitectura multinodo de Hadoop brinda mejor tiempo de respuesta en comparación a una arquitectura de nodo singular.
- La ejecución de las pruebas permitió determinar que la herramienta RHadoop se encuentra correctamente integrada al procesamiento de la arquitectura multinodo de Hadoop, así mismo que se puede recuperar el resultado para la generación de gráficas que permiten la correcta interpretación de los resultados.
- La prueba de sistema y de rendimiento permitió comprobar el tiempo de respuesta de la ejecución de cada uno de los scripts, así mismo se realizó una comparación con los resultados que se obtiene de la ejecución en una arquitectura de nodo singular, permitiendo identificar que la arquitectura multinodo de Hadoop brinda mejores tiempos de respuesta.
- Una vez que se comprobó el correcto funcionamiento de la arquitectura y de las herramientas utilizadas para el procesamiento y obtención de los resultados, se procedió a desarrollar un prototipo que permite la presentación de los resultados a un interesado, el prototipo se desarrollo en el lenguaje de programación php, y a este se le ejecutó una serie de pruebas, una prueba de interfaz de usuario donde se comprobó que la interfaz desarrollada cumple con todas las características de desarrollo, y una prueba de calidad de software que se hizo en la herramienta SonarQube en donde se comprueba la calidad del mismo y se obtuvo buenas calificaciones.

5.5. Comentarios Finales.

El estudio de los datos generados por las mismas instituciones es de suma importancia porque permite conocer el estado pasado o actual de la misma y así determinar o identificar algún error que conllevo a la obtención de dichos resultados, es así que hoy en día muchas instituciones han aprendido de sus errores y no los han vuelto a cometer gracias al estudio de sus datos, además esto les ha servido para la

creación de nuevos servicios o estrategias de negocio que les permiten generar más ganancias y sacar ventaja a su competencia.

El uso de la metodología CRISP-DM en el desarrollo del análisis y visualización de los datos que se realizó en este trabajo ha permitido conocer los tipos de datos que genera la organización mediante la utilización de los diferentes sistemas con los que cuenta, es así que en este trabajo se utilizaron datos con los cuales se pudo identificar y visualizar los diferentes resultados. Este análisis además permitió conocer toda la información relevante de acuerdo a los datos obtenidos. Para la presentación de los resultados se trabajó en el desarrollo de un prototipo que ejecuta un script de R en el cual internamente funciona la infraestructura multinodo de Hadoop con la cual se realiza el procesamiento de la data.

El lado positivo de haber desarrollado este análisis y visualización de la data fue la utilización de esta nueva tecnología que es Hadoop y hoy en día es una de las herramientas que más está creciendo en el análisis de datos debido a las diferentes herramientas que proporciona como lo es Hive, Spark, Oozie, Pig, etc. En el desarrollo de este trabajo para el análisis se utilizó R que no es una herramienta desarrollada por el equipo de Hadoop, sino que es una herramienta que permite el análisis y exploración de la data en el mismo programa y se pueden diseñar una serie de visualizaciones de los datos trabajados, para aprovechar esta ventaja que ofrece R se realizó la integración con Hadoop gracias a las librerías producidas por el equipo de Hadoop, es así que se llevó a cabo el análisis, procesamiento y visualización gracias a las ventajas que poseen estas dos herramientas globales.

CONCLUSIONES

Al haber finalizado el presente trabajo y habiendo cumplido los objetivos planteados se concluye que:

- El término de Big Data encierra una serie de definiciones que en fin se determina como un gran volumen de datos que son generados por diferentes fuentes ya sean actividades, dispositivos, tecnologías, etc. y esto conlleva a encontrarse datos que son de diferente tipo como estructurados, semiestructurados y no estructurados.
- El estudio de Big Data hoy en día es de suma importancia ya que al ser procesada y analizada por las herramientas apropiadas permite a las instituciones y organizaciones realizar la correcta toma de decisiones y creación de nuevas estrategias de negocio basándose en los resultados obtenidos del estudio de los datos generados por ellos mismos, lo que a su vez se genera en ventaja contra sus competidores.
- El estudio y análisis de Big Data se realiza en base a la utilización de tecnologías y arquitecturas diseñadas que permiten conseguir un mejor rendimiento de procesamiento de grandes volúmenes de datos. Para la obtención de los beneficios como velocidad y veracidad que conlleva el estudio de Big Data depende mucho de los equipos tecnológicos que permitan su análisis y procesamiento, además de la calidad del personal con la que se cuenta ya que ellos son los encargados de realizar una correcta interpretación de los resultados y esto se transforma en valor para la organización. Es así como en el presente trabajo de titulación se implementa una arquitectura multinodo de Hadoop la cual permite el multiprocesamiento de los datos y con este resultado obtenido se trabaja en R para generar las diferentes visualizaciones que sirven de utilidad a las personas encargadas de revisar los resultados del estudio del volumen de datos y realicen la corrección o creación de nuevos servicios en base a la toma de decisiones.
- Rhadoop es la mejor herramienta de procesamiento y visualización de Big Data, R aporta su poder de análisis, exploración y visualización, en cambio Hadoop ofrece el poder de multiprocesamiento de los datos, la integración entre estas dos herramientas se logra a la utilización en R de las librerías rmr2 que proporciona la funcionalidad de MapReduce y la librería rhdfs que proporciona la administración de los archivos de Hadoop. Juntas estas herramientas permiten que la ejecución de este trabajo tenga un final satisfactorio al entender su funcionamiento y generación de resultados en base

al estudio, análisis, procesamiento y visualización del volumen de datos con el que se trabaja, los resultados de las visualizaciones obtenidas permitirán la correcta toma de decisiones a las diferentes personas encargadas o gerentes institucionales.

- En el desarrollo de este trabajo se realiza la instalación y configuración de una arquitectura multinodo de Hadoop, con el fin de aprovechar todas sus características de funcionamiento, esta implementación e integración fue un laboriosa, pero se obtuvo una gran satisfacción de haber cumplido con éxito al comprobar el funcionamiento de esta infraestructura que permite el multiprocesamiento del volumen de datos con el que se trabaja en el presente trabajo de titulación, el resultado de este procesamiento es utilizado en R para realizar las diferentes operaciones como análisis y generación de visualizaciones que sirven de apoyo a la toma de decisiones.
- Al finalizar la instalación, configuración e integración de las herramientas en los tres equipos que se utilizan para el desarrollo de este trabajo de titulación se empieza a realizar las primeras pruebas de funcionamiento de la integración, a continuación, se procede a cargar el volumen de datos y realizar los primeros pasos del análisis y procesamiento en RHadoop, como resultado de este análisis y procesamiento se conoce la estructura del volumen de datos y que resultados se puede obtener, es así que posteriormente se divide en subgrupos el volumen de datos para obtener las gráficas como calificaciones finales, número de estudiantes matriculados, estado de registro y generación de un mapa de estudiantes matriculados a nivel nacional por provincia, resultados que demuestran la veracidad del análisis y procesamiento del volumen de datos, las visualizaciones generadas son fácilmente entendibles para cualquier persona que desee conocer el resultado así mismo para las autoridades o gerentes institucionales encargados de generar nuevas estrategias de negocio y realizar la toma de decisiones.
- Con la implementación de la metodología de análisis de datos CRISP-DM, se obtiene los beneficios de llevar una planificación eficiente y ordenada del análisis de los datos lo que se traduce en obtener resultados óptimos que permiten la correcta interpretación de los resultados.
- El diseño y desarrollo del prototipo se lo realizó con el fin de que su usabilidad sea eficiente y entendible, permitiendo una visualización agradable e identificación de los resultados de forma clara y precisa, los resultados presentados en el prototipo son resultado de la integración entre las herramientas, el análisis, procesamiento y visualización de la data realizado en

la herramienta RHadoop, se espera que estos resultados sean de ayuda para la creación de nuevas estrategias de negocio y a la toma de decisiones dentro de la universidad.

- La fase de pruebas permitió comprobar e identificar el correcto funcionamiento de la arquitectura multinodo de Hadoop y realizar una comparación con una arquitectura de nodo singular, con lo cual se pudo verificar a través de pruebas de rendimiento de sistema que una arquitectura multinodo brinda mayor velocidad de procesamiento lo que se traduce en tiempo de ganancia para la obtención de resultados.
- En base a los resultados obtenidos en el desarrollo del presente trabajo de titulación, queda el precedente para los gerentes de las diferentes instituciones u organizaciones en que se puede trabajar en proyectos donde se pueda acceder y conocer los resultados del estudio de su volumen de datos de una forma automatizada y dinámica mediante la utilización de esta nueva tecnología que es RHadoop.

RECOMENDACIONES

Tras haber con concluido el presente trabajo y a la experiencia adquirida en el desarrollo del mismo, se presenta las siguientes recomendaciones a tomar en cuenta como punto de partida para futuros trabajos de análisis y visualización de datos mediante RHadoop o en desarrollo de proyectos relacionados con Big Data.

- Para realizar trabajos de análisis y procesamiento de Big Data se recomienda que los equipos a utilizar tengan buenas características como un buen procesador, memoria interna y memoria RAM, esto ayudará a la velocidad de obtención de resultado.
- Para realizar trabajos de análisis de datos se recomienda utilizar una de las metodologías descritas en el presente trabajo de titulación, ya que cada una de las metodologías son las más utilizadas en el desarrollo de este tipo de trabajos y cada una de las fases brinda resultados que cumplen con estándares de calidad para proseguir con cada una de las siguientes fases.
- Como aporte personal la infraestructura multinodo de Hadoop, RHadoop y el prototipo desarrollado para visualizar los resultados queda implementado en el laboratorio de datos UTPL, en donde cualquier persona interesada pueda utilizar la arquitectura, visualizar los resultados del prototipo y pueda trabajar con un nuevo conjunto de datos e implementar las diferentes herramientas que ofrece el ecosistema Hadoop y así trabajar en nuevos proyectos de análisis procesamiento y visualización con RHadoop.
- Se recomienda llevar una correcta planificación y aplicación de un plan de pruebas ya que permite identificar si existe algún tipo de falla en el funcionamiento del sistema o en una arquitectura, así se asegura que el resultado a obtener cumpla con los atributos de calidad.
- Se recomienda trabajar con el sistema operativo Ubuntu 14.04, con la versión de Hadoop 2.6.5 y en versión de R se puede trabajar con cualquier versión ya que es software libre y no se tiene ningún problema al integrar con Hadoop, si cumple con lo antes mencionado logrará una correcta integración y funcionamiento de las herramientas para proseguir con el análisis y procesamiento de Big Data.

BIBLIOGRAFÍA

Ames, A. J., Abbey, R., & Thompson, W. (2013). Big Data Analytics. *SAS Institute Inc., Cary, NC*, 1–15. Retrieved from https://support.sas.com/content/dam/SAS/support/en/technical-papers/data-text-mining/Benchmark_R_Mahout_SAS.pdf

Bagwari, N., & Kumar, O. (2017). Indexing optimizations on Hadoop, 1–7. Retrieved from https://doi.org/10.1109/CIACT.2017.7977360

BBVA Innovation Center. (2013). Big Data Es hora de generar valor de negocio con los datos. *6*, 18. Retrieved from https://www.centrodeinnovacionbbva.com/documentos/pdfs/bigdata_spanish.pdf

Bhupathiraju, V., & Ravuri, R. P. (2014). The dawn of Big Data - Hbase. *Proceedings of the 2014 Conference on IT in Business, Industry and Government: An International Conference by CSI on Big Data, CSIBIG 2014*, 0–3. https://doi.org/10.1109/CSIBIG.2014.7056952

Borthakur, D. (2008). HDFS architecture guide. *Hadoop Apache Project Http://hadoop Apache …*, 1–13. Retrieved from http://archive.cloudera.com/cdh/3/hadoop-0.20.2-cdh3u6/hdfs_design.pdf%5Cnpapers3://publication/uuid/BE03DF70-D0C1-441E-A65F-1888C84992D6

Brynjolfsson, E., & McAfee, A. (2012). Big Data: The Management, (October), 1–14. Retrieved from http://tarjomefa.com/wp-content/uploads/2017/04/6539-English-TarjomeFa-1.pdf

Chen, Z. (2017). Processing and Analysis of Seismic data in Hadoop Platform, 1–5. Retrieved from http://ieeexplore.ieee.org/document/7977288/

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2007). Map-Reduce for Machine Learning on Multicore. *Advances in Neural Information Processing Systems 19*, 281–288. https://doi.org/10.1234/12345678

Climent, M., & Mallol, E. (2015). Así cambiará tu vida el Big Data en 11 ejemplos prácticos | Innovadores | EL MUNDO. Retrieved February 1, 2017, from http://www.elmundo.es/economia/2015/05/22/555ef33422601dba5d8b4577.pdf

Condés, O. (2016). Mejores herramientas de ayuda para visualización de datos. Retrieved February 1, 2017, from http://www.ticbeat.com/empresa-b2b/herramientas-de-visualizacion-datos/

Corrales, D. C., Ledezma, A., & Corrales, J. C. (2015). A Conceptual Framework for

Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal. *Journal of Computers*, *10*(6), 396–405. https://doi.org/10.17706/jcp.10.6.396-405

Dean, B. Y. J., & Ghemawat, S. (2010). MapReduce: a flexible data processing tool. *Communications of the ACM*, *53*(1), 72–77. https://doi.org/10.1145/1629175.1629198

Dolák, O. (2011). Big data, 561–566. Retrieved from https://www.systemonline.cz/clanky/big-data.htm

Douglas, C., Lowe, J., Malley, O. O., & Reed, B. (2013). Apache Hadoop YARN : Yet Another Resource Negotiator. Retrieved from https://www.cse.iitb.ac.in/synerg/lib/exe/fetch.php?media=public:students:golharj:reviewonapachehadoopyarnyetanotherresourcenegotiator.pdf

En qué consiste big data analytics y cómo beneficia a tu empresa. (2016). Retrieved December 28, 2016, from http://www.lantares.com/blog/en-que-consiste-big-data-analytics-y-como-beneficia-a-tu-empresa

Francisco, E. D. R. (2015). Big Data: https://doi.org/10.1007/978-81-322-2494-5

Galicia, C. y L. (n.d.). Conceptos Básicos BigData. Retrieved from http://www.trc.es/pdf/descargas/big_data.pdf

Gill, S. K., Nguyen, P., & Koren, G. (2009). Adherence and tolerability of iron-containing prenatal multivitamins in pregnant women with pre-existing gastrointestinal conditions. *Journal of Obstetrics and Gynaecology*, *29*(7), 594–598. https://doi.org/10.1080/01443610903114527

Hu, H., Wen, Y., & Li, X. (2014). A Framework for Big Data Analytics as a Scalable Systems. *IEEE Access*, *2*, 652–687. https://doi.org/10.1109/ACCESS.2014.2332453

Kala Karun, A., & Chitharanjan, K. (2013). A review on hadoop - HDFS infrastructure extensions. *2013 IEEE Conference on Information and Communication Technologies, ICT 2013*, (Ict), 132–137. https://doi.org/10.1109/CICT.2013.6558077

Mallinger, M. (2015). Big Data Decision Making: Is There Room for Intuition in the Era of Big Data ? Is There Room for Intuition in the Era of Big Data? *Graziadio Business Report ·*, *18*(2). https://www.researchgate.net/publication/283883860_Big_Data_Decision_Making_Is_There_Room_for_Intuition_in_the_Era_of_Big_Data

Patel, D. D., & Singh, K. R. (2017). Genome Sequencing using MapReduce and Hadoop – A Technical Review, (Icimia), 544–547. https://www.irjet.net/archives/V4/i4/IRJET-V4I4359.pdf

Patil, P. P., & Phatak, M. V. (2014). International Journal of Emerging Technology and Advanced Engineering, (October). http://www.ijetae.com/files/Volume4Issue5/IJETAE_0514_58.pdf

Pico, R. (2015). Big data: cómo las grandes empresas lo utilizan en su estrategia de producto. Retrieved February 1, 2017, from http://www.puromarketing.com/12/23633/big-data-como-grandes-empresas-utilizan-estrategia-producto.pdf

Puyol Moreno, J. (2014). UNA APROXIMACIÓN A BIG DATA, 471–506.

SAS. (2015). Big data analytics: What it is and why it matters | SAS. Retrieved February 1, 2017, from http://www.sas.com/en_us/insights/analytics/big-data-analytics.pdf

Sathi, A. (2012). *Big Data Analytics*. *MC Press Online*. https://doi.org/10.1017/CBO9781107415324.004

Schroeck, Michael; Shockley, Rebecca; Smart, J. (2012). Analytics: el uso de big data en el mundo real. *IBM. Informe Ejecutivo*, 22. Retrieved from http://www-05.ibm.com/services/es/bcs/pdf/Big_Data_ES.PDF

Sethia, D., Sheoran, S., & Saran, H. (2017). Optimized MapFile based Storage of Small files in Hadoop. *CCGrid '17 Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 906–912. https://doi.org/10.1109/CCGRID.2017.83

Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD , CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217–222. Retrieved from http://www.ijisr.issr-journals.org/

Shulyak, A. C., & John, L. K. (2016). Identifying performance bottlenecks in Hive: Use of processor counters. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2109–2114. https://doi.org/10.1109/BigData.2016.7840838

Shvachko, K. (2010). HDFS Scalability: The limits to growth. *Login*, 6–16. Retrieved from http://c59951.r51.cf2.rackcdn.com/5424-1908-shvachko.pdf

Shvachko, K. (2010). The Hadoop Distributed File System. *IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. https://doi.org/10.1109/MSST.2010.5496972

Singh, S., & Singh, N. (2012). Big Data analytics. *2012 International Conference on Communication, Information {&} Computing Technology (ICCICT)*, 1–4. https://doi.org/10.1109/ICCICT.2012.6398180

Suriol, A. G. (2014). La creación de valor en las empresas a través del Big Data, 46. http://diposit.ub.edu/dspace/bitstream/2445/67546/1/TFG-ADE-Galimany-Aleix-juliol15.pdf

Worms, D. (2012). Hadoop y R con RHadoop - Adaltas. Retrieved February 1, 2017, from http://www.adaltas.com/blog/2012/05/19/hadoop-and-r-is-rhadoop/

Wu, X., Zhu, X., Wu, G., & Ding, W. (2013). Data Mining with Big Data, (Ibm 2012). Retrieved from http://ieeexplore.ieee.org/abstract/document/6547630/

Yao, Y., Wang, J., Sheng, B., Lin, J., & Mi, N. (2014). HaSTE: Hadoop YARN scheduling based on task-dependency and resource-demand. *IEEE International Conference on Cloud Computing, CLOUD*, 184–191. https://doi.org/10.1109/CLOUD.2014.34

Yu-Wei, & Chiu, D. (2015). Los análisis de datos (R y Hadoop) | PACKT Libros. Retrieved February 1, 2017, from https://www.packtpub.com/books/content/big-data-analysis-r-and-hadoop

Yun, Z., Weihua, L., & Yang, C. (2014). Applying Balanced ScoreCard Strategic Performance Management to CRISP-DM. *International Conference on Information Science, Electronics and Electrical Engineering, ISEEE 2014*. http://www.scielo.br/pdf/bar/v6n4/v6n4a06.pdf

Zakir, J., Seymour, T., & Berg, K. (2015). Big Data Analytics. *Issues in Information Systems*, *16*(2), 81–90. Retrieved from http://www.iacis.org/iis/2015/2_iis_2015_81-90.pdf

Zhou, Y. G. and J. R. and X. (2013). iShuffle: Improving Hadoop Performance with Shuffle-on-Write. *10th International Conference on Autonomic Computing, ICAC'13, San Jose, CA, USA, June 26-28, 2013*, 107–117. Retrieved from https://www.usenix.org/conference/icac13/technical-sessions/presentation/guo

ANEXOS

Anexo 1: Instalación de Hadoop

Pre requisitos de Instalación.

1. Tener instalado Ubuntu 14.04.
2. Tener instalada la versión de java por defecto, en este caso la versión instalada es 1.7.0_151.

Configuración de Linux antes de la instalación de Hadoop

En este anexo se explica detenidamente el procedimiento para configurar un clúster Hadoop de nodo único en Ubuntu 14.04. Se espera que conozca los comandos básicos de UNIX y los comandos del editor nano.

Es necesario ejecutar los comandos que están marcados sólo en color rojo.

Vamos a configurar el nodo único Hadoop clúster utilizando un usuario dedicado Hadoop llamado “hadoop”.

1. **Inicie sesión como root.**

   $ sudo su

   # **whoami** - este comando debería dar el usuario raíz.

2. **Agregar un usuario del sistema Hadoop dedicado llamado “hadoop”.**

   Se utiliza un usuario dedicado para ejecutar Hadoop.

3. **Agregue el usuario “hadoop” a la lista de sudoers o super usuarios para que el usuario “hadoop” pueda hacer tareas de administrador.**

   $ sudo visudo

   Agregue una línea bajo #Permita que el miembro del grupo sudo ejecute cualquier comando en cualquier lugar del formato.

   hadoop ALL = (ALL) ALL

   Pulse ctrl + x, excriba Y luego enter

   Esto agregará el usuario “hadoop” a su máquina local.

4. **Configuración de SSH.**

   Hadoop requiere el acceso SSH para administrar sus nodos, es decir, máquinas remotas más su máquina local si desea usar Hadoop en él. Para nuestra configuración de nodo único de Hadoop, necesitamos configurar el acceso SSH a localhost para el usuario de “hadoop”.

   En primer lugar, tenemos que generar una clave SSH para el usuario “hadoop”.

   hadoop@ubuntu:~$ sudo apt-get install openssh-server

   Ingrese su password, luego Y para continuar.

5. **Genere SSH para la comunicación.**

   hadoop@ubuntu:~$ ssh-keygen

   Simplemente presione Enter para los mensajes que se presenten.

Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
9b:82:ea:58:b4:e0:35:d7:ff:19:66:a6:ef:ae:0e:d2hduser@localhost The key's randomart image is:
[...snipp...]

hadoop@ubuntu: ~$

6. **Copie la clave pública al archivo de la clave autorizada y edite el permiso.**

Ahora copia la clave pública en el archivo authorized_keys, para que ssh no necesite contraseñas cada vez, esto se realiza con el siguiente comando:

hadoop@ubuntu:~$ cat ~ /.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Cambie los permisos de authorized_keys para obtener todos los permisos para el usuario “hadoop”, con el siguiente comando.

hadoop@ubuntu:~$ chmod 700 ~ /.ssh/authorized_keys

7. **Inicie SSH.**

Si ssh no se está ejecutando, ejecútelo dando el siguiente comando:

hadoop@ubuntu:~$ sudo /etc/init.d/ssh restart

Introduzca su contraseña.

8. **Deshabilitar IPv6.**

Hadoop e IPv6 no están de acuerdo en el significado de la dirección 0.0.0.0, por lo que es aconsejable desactivar IPv6 añadiendo las siguientes líneas al final en el archivo de configuración /etc/sysctl.conf

hadoop@ubuntu:~ $ sudo vim /etc/sysctl.conf

Introduzca su contraseña y añada las siguientes líneas al final del archivo:

 # disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

El archivo de configuración quedaría de la siguiente manera:

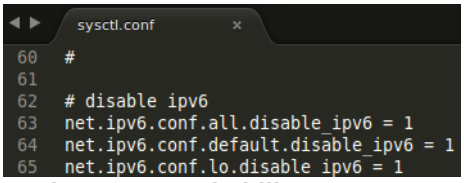


Figura 72 - Dehabilitar IPv6
Fuente: Elaboración Propia

9. **Compruebe si IPv6 está deshabilitado.**

   Para comprobar que IPv6 está deshabilitado se debe ingresar el siguiente comando:

   hadoop@ubuntu:~$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6

   El mensaje mostrado debe ser 1, lo que significa que IPV6 está realmente deshabilitado. Si la respuesta es 0 debe reiniciar el equipo para que los cambios surjan efecto.

Instalación de Hadoop.

1. **Descargar Hadoop.**

   Para la realización de este trabajo se utiliza la versión Hadoop 2.6.5, se descarga desde el siguiente enlace:

   http://www-us.apache.org/dist/hadoop/common/

Figura 73 - Versiones de Hadoop
Fuente: Elaboración Propia

Descargue la versión hadoop-2.6.5.tar.gz y guárdelo en Descargas.

Figura 74 - Descarga Hadoop 2.6.5
Fuente: Elaboración Propia

2. **Mueva el archivo zip a /usr/local/**

   Ingrese a una terminal y escriba los siguientes comandos:

   $ sudo mv Descargas/hadoop-2.7.3.tar.gz /usr/local/
   Enter password:
   $ cd /usr/local

sudo tar -xvf hadoop-2.7.3.tar.gz
sudo rm hadoop-2.7.3.tar.gz
sudo ln -s hadoop-2.7.3 hadoop
sudo chown -R hadoop:hadoop hadoop-2.7.3
sudo chmod 777 hadoop-2.7.3

3. **Edite hadoop-env.sh y configure Java.**

Para editar el archivo hadoop-env.sh se debe ingresar a la siguiente ruta /usr/local/hadoop/etc/hadoop/hadoop-env.sh en la cual se debe eliminar la siguiente línea:

export JAVA_HOME = $ {JAVA_HOME}

Para abrir el archive a modificar se realiza con el siguiente comando:

$ sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh

El archivo de configuración quedaría de la siguiente manera:

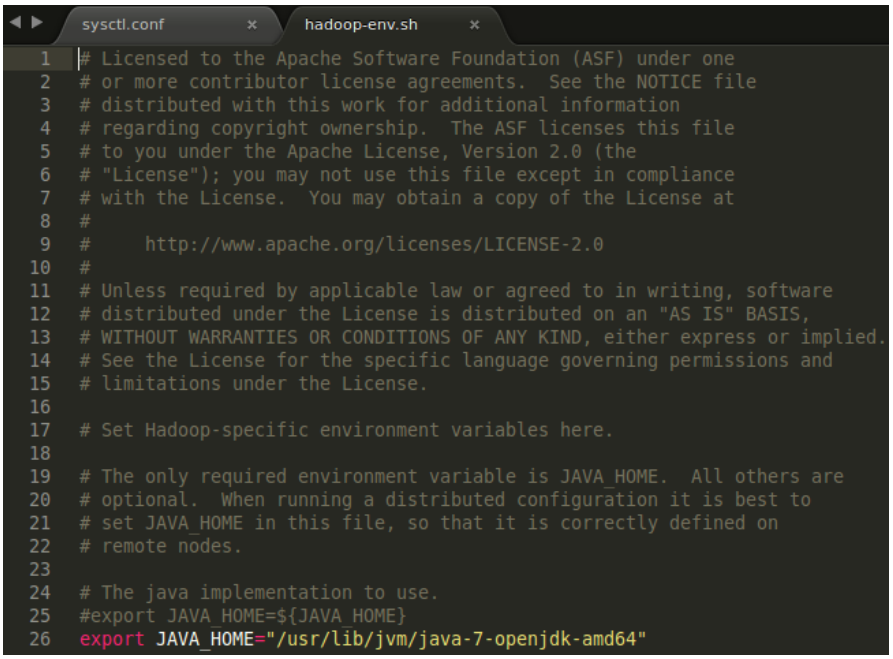


Figura 75 - Configuración Java hadoop-env.sh
Fuente: Elaboración Propia

4. **Configurar el archivo bashrc.**

Agregue las siguientes líneas al final del archivo $HOME /.bashrc del usuario "hadoop". Abrir el archivo con el siguiente comando:

$ sudo nano ~/.bashrc

Añadir lo siguiente al final:

Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_PREFIX=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}

export HADOOP_YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

Se necesita cerrar el terminal y abrir uno nuevo para que los cambios hagan efecto.

El archivo de configuración quedaría de la siguiente manera:

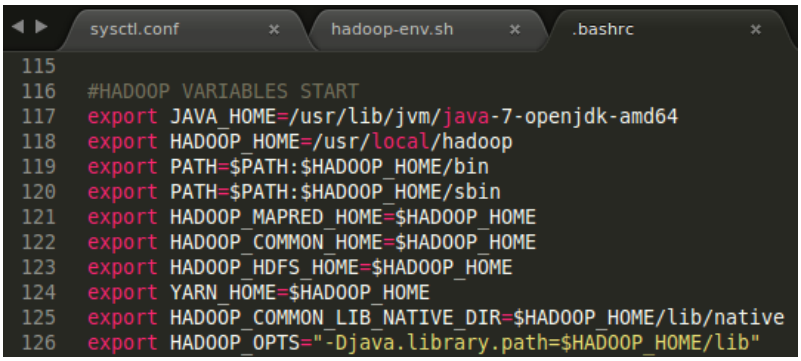


Figura 76 - Editar archivo bashrc
Fuente: Elaboración Propia

5. **Configurar yarn-site.xml.**

Se abre el archivo de configuración con el siguiente comando:

$ sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml

Añadir lo siguiente entre <configuration> …….. </configuration>

```
<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>

<property>
        <name>yarn.nodemanager.aux-
services.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

El archivo de configuración quedaría de la siguiente manera:

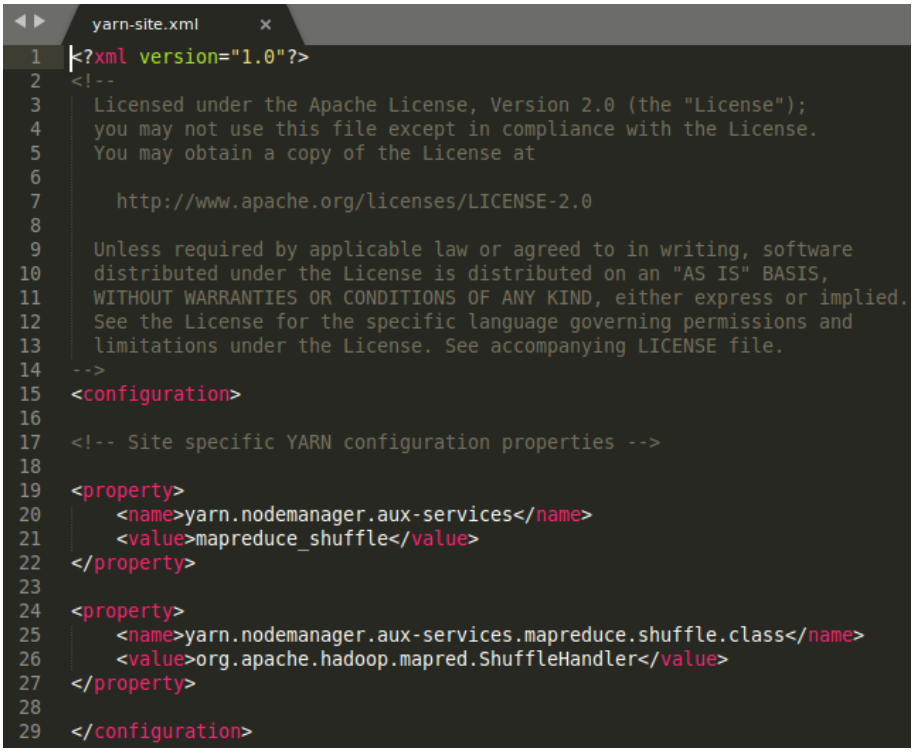


Figura 77 - Configuración yarn-site.xml
Fuente: Elaboración Propia

6. **Configurar core-site.xml.**

Se abre el archivo de configuración con el siguiente comando:

$ sudo vim /usr/local/hadoop/etc/hadoop/core-site.xml

Añadir lo siguiente entre <configuration> …….. </configuration>

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
</property>

<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
</property>
```

El archivo de configuración quedaría de la siguiente manera:

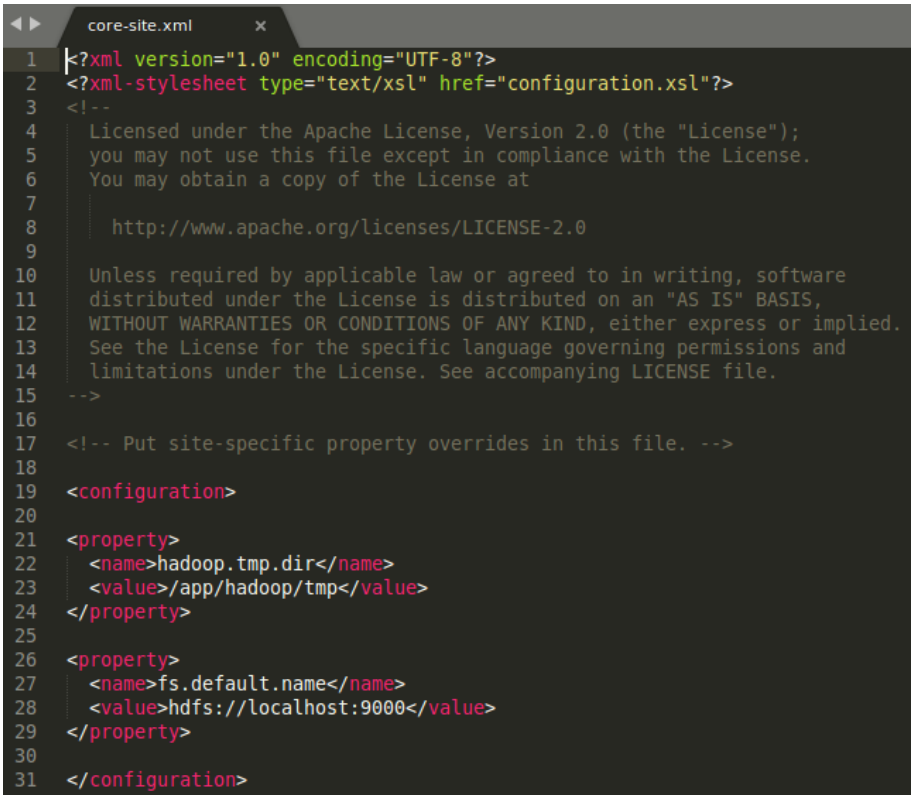


Figura 78 - Configuración core-site.xml
Fuente: Elaboración Propia

7. **Crear la carpeta temp y proporcione los permisos apropiados.**

   $ sudo mkdir -p /app/hadoop/tmp

   $ sudo chown hadoop:hadoop -R /app/hadoop/tmp

   $ sudo chmod 750 /app/hadoop/tmp

8. **Crear el archivo mapred-site.xml desde el archivo de configuración mapred-site.xml.template.**

   Para realizar la creación de este archivo se escribe lo siguiente en el terminal:

   $ sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml

   Añadir lo siguiente entre <configuration> …….. </configuration>

```
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
</property>

<property>
        <name>mapreduce.jobhistory.address</name>
        <value>localhost:10020</value>
</property>
```

   El archivo de configuración quedaría de la siguiente manera:

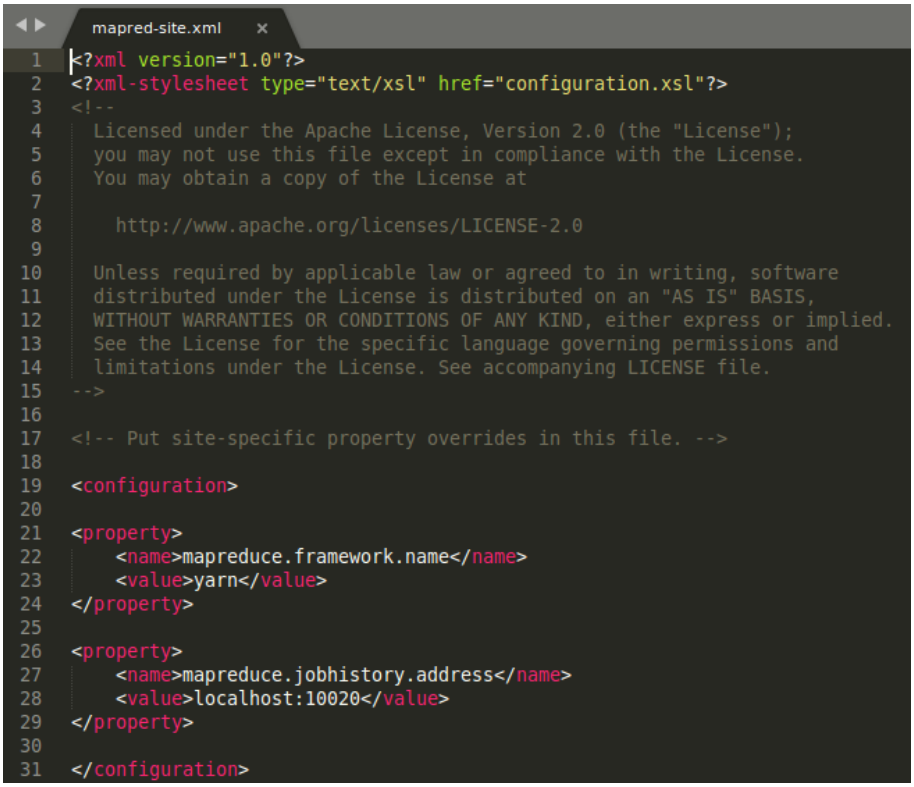


Figura 79 - Configuración mapred-site.xml
Fuente: Elaboración Propia

9. **Crear un directorio temporal que se utilizará como ubicación base para DFS.**

   Se crea el directorio y se establecen los propietarios y permisos necesarios con los siguientes comandos:

   $ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode

   $ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode

   $ sudo chown hadoop:hadoop -R /usr/local/hadoop_tmp/

10. **Configurar el archivo hdfs-site.xml.**

    Se abre el archivo de configuración con el siguiente comando:

    $ sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml

    Añadir lo siguiente entre <configuration> …….. </configuration>

```
<property>
        <name>dfs.replication</name>
        <value>1</value>
</property>

<property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>

<property>
        <name>dfs.datanode.data.dir</name>
```

<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>

El archivo de configuración quedaría de la siguiente manera:

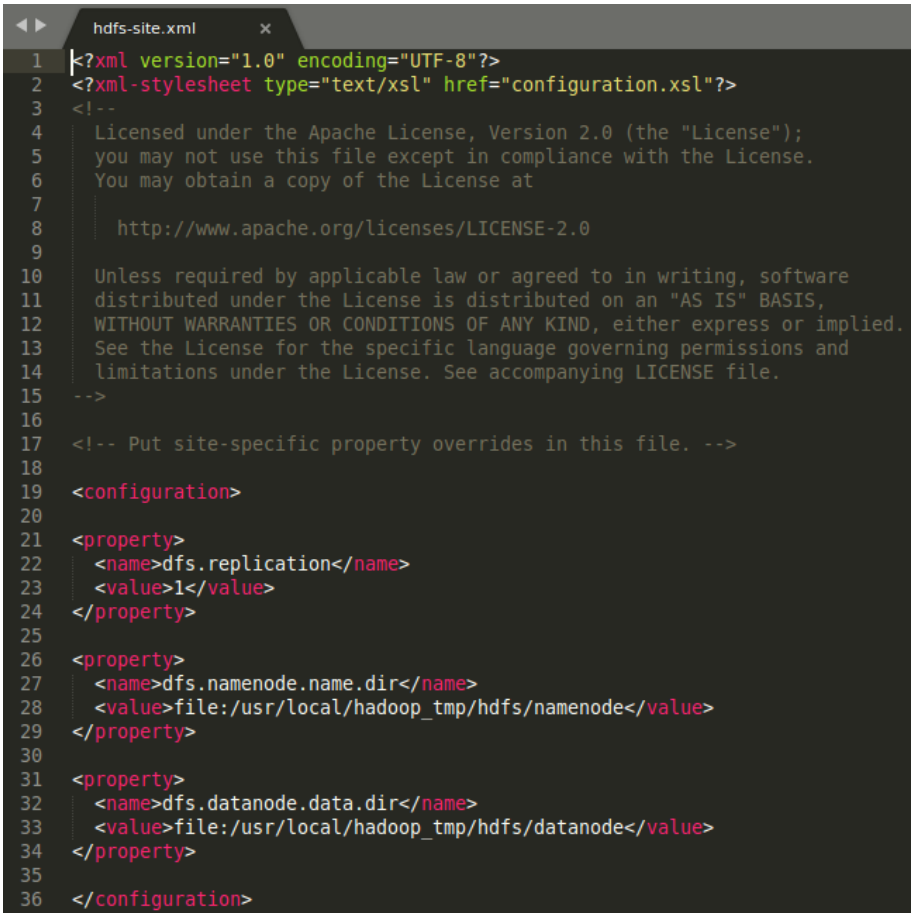


Figura 80 - Configuración hdfs-site.xml
Fuente: Elaboración Propia

11. Formatear namenode.

Abrir un nuevo terminal sino el comando hadoop no funcionará. Se formatea el clúster hdfs con el siguiente comando:

$ hadoop namenode –format

12. Inicie el clúster single-node

Pruebe el clúster ejecutando los siguientes comandos:

$ start-dfs.sh --inicia NN, SNN, DN – Digite yes si se ha solicitado algo

$ start-yarn.sh --inicia NodeManager, ResourceManager

13. Compruebe si todo el domino hadoop se está ejecutando o no.

Esto se realiza ejecutando el siguiente comando:

$ jps

Se debe obtener la siguiente respuesta:

4912 NameNode
5361 ResourceManager
5780 Jps
5209 SecondaryNameNode
5485 NodeManager
5251 DataNode
3979 JobHistoryServer

14. **Comprobar si la carpeta de inicio se ha creado o no en hdfs.**

Para comprobar se escribe el siguiente comando:

$ hadoop fs -ls

Si se obtiene el siguiente mensaje:

16/06/23 13:47:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

ls: `.': No such file or directory

Significa que su directorio principal hadoop no se ha creado correctamente. Escriba el siguiente comando en el terminal:

$ hadoop fs -mkdir -p /user/hadoop **(Deprecated)**

$ hdfs dfs -mkdir -p /user/hadoop **(Use this)**

Ahora no se debería mostrar el error con el siguiente comando. Por primera vez no obtendrá ninguna salida ya que la carpeta de inicio de hdfs está vacía.

$ hadoop fs -ls

15. **Comprobar si se puede acceder a Hadoop mediante el navegador pulsando las siguientes URL:**

localhost:50070

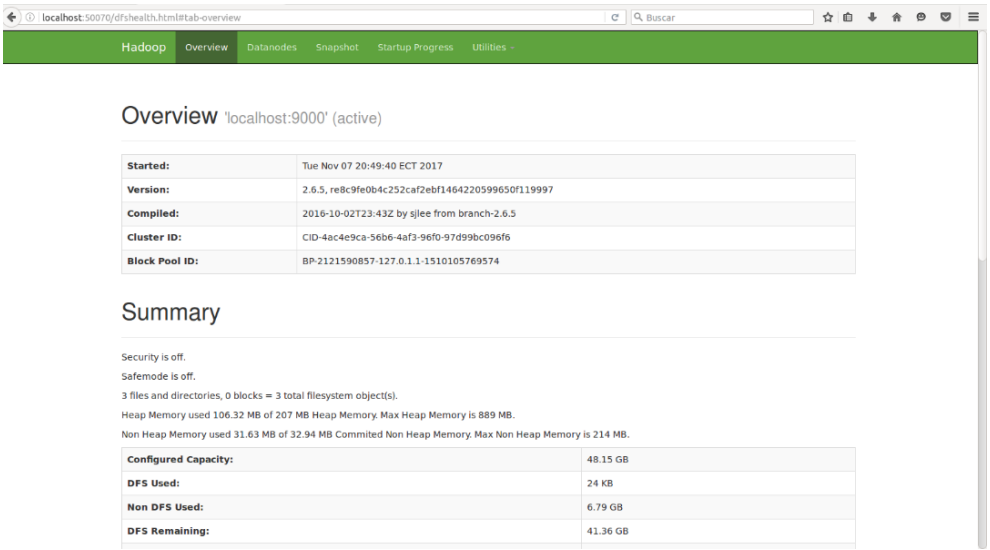


Figura 81 - Comprobar Hadoop Singlenode1
Fuente: Elaboración Propia

localhost:8088

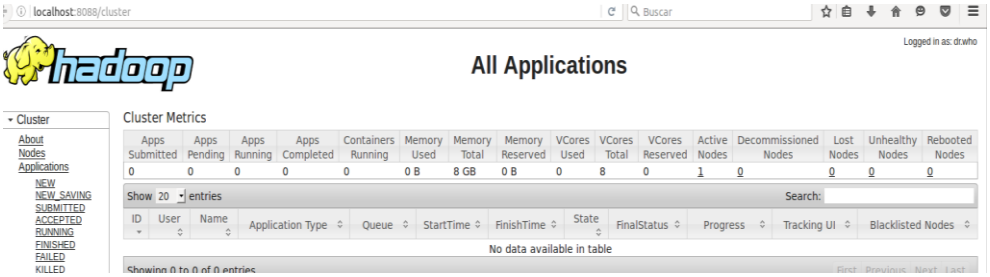


Figura 82 - Comprobar Hadoop Singlenode2
Fuente: Elaboración Propia

Anexo 2: Instalación de Hadoop Multi Nodo

Pre requisitos de Instalación.

1. Tener instalado Ubuntu 14.04.
2. Tener instalada la versión de java por defecto, en este caso la versión instalada es 1.7.0_151.
3. Tener instalado y configurado Hadoop como nodo singular en cada una de las máquinas, en este caso se utilizarán 3 máquinas.

Configuración de las máquinas.

En este anexo se explica detenidamente el procedimiento para configurar un clúster Hadoop de multi nodo en Ubuntu 14.04. Se espera que conozca los comandos básicos de UNIX y los comandos del editor nano.

Es necesario ejecutar los comandos que están marcados sólo en color rojo.

Vamos a configurar el nodo único Hadoop clúster utilizando un usuario dedicado Hadoop llamado “hadoop”.

1. **Asignar dirección IP a cada una de las máquinas.**

   Dirigirse a Editar Conexiones.

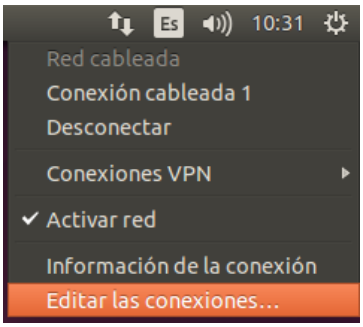


Figura 83 - Editar Conexión de Red1
Fuente: Elaboración Propia

   Seleccionar la conexión a editar, en este caso conexión cableada 1, clic en editar.

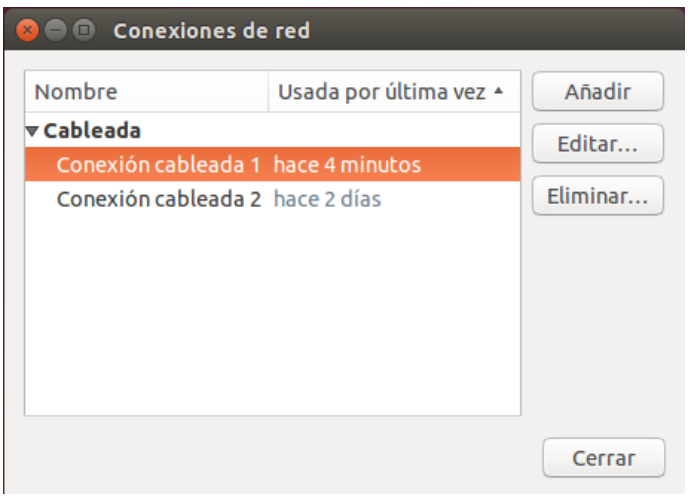


Figura 84 - Editar Conexión de Red2
Fuente: Elaboración Propia

Seleccionar añadir y posteriormente asignar una dirección IP como en la imagen, para guardar los cambios clic en Guardar.

Figura 85 - Editar Conexión de Red3
Fuente: Elaboración Propia

Para verificar que la dirección este asignada abrir un terminal y digitar el comando ifconfig, debería mostrar la dirección asignada anteriormente.

Asignar de la siguiente manera las direcciones a cada una de las maquinas a utilizar:

Master 192.168.220.2

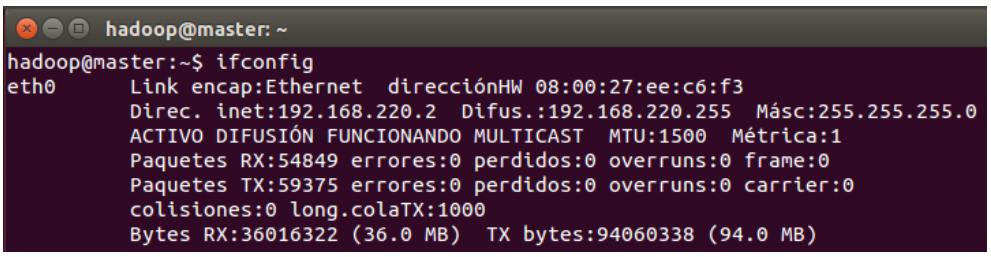


Figura 86 - Comprobar Ip Master
Fuente: Elaboración Propia

Slave1 192.168.220.3

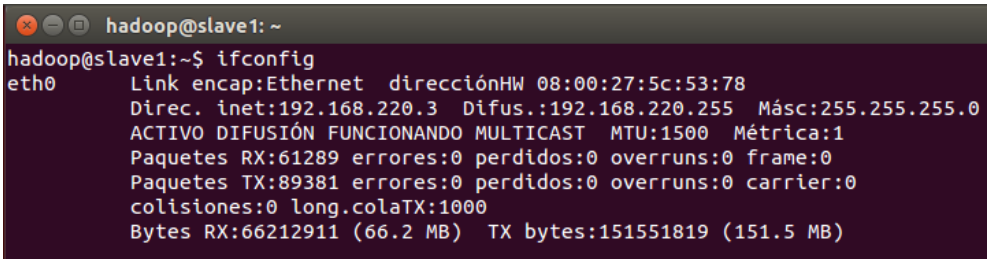


Figura 87 - Comprobar Ip Slave1
Fuente: Elaboración Propia

Slave2 192.168.220.4

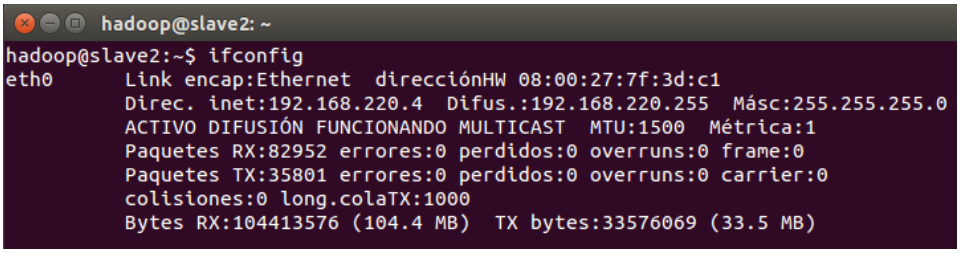


Figura 88 - Comprobar Ip Slave2
Fuente: Elaboración Propia

2. **Verificar que exista comunicación entre las máquinas.**

Para realizar la comprobación de comunicación digitar lo siguiente en cada una de las máquinas.

Desde el master realizar lo siguiente:

hadoop@ubuntu:~$ ping 192.168.220.3 //Master ping a slave1

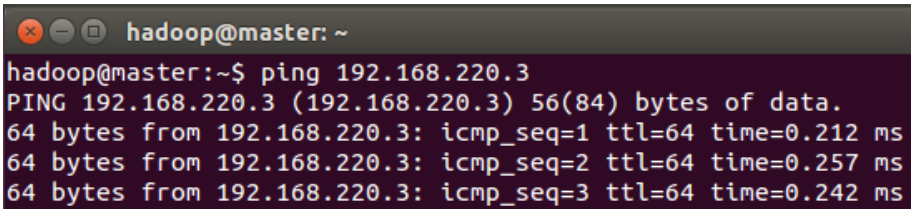


Figura 89 - Ping Master a Slave1
Fuente: Elaboración Propia

hadoop@ubuntu:~$ ping 192.168.220.4 //Master ping a slave2

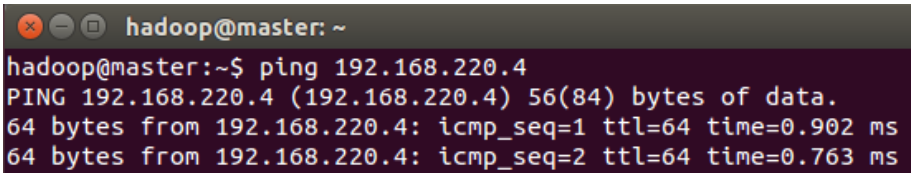


Figura 90 - Ping Master a Slave2
Fuente: Elaboración Propia

Desde el slave1 realizar lo siguiente:

hadoop@ubuntu:~$ ping 192.168.220.2 //Slave1 ping a master

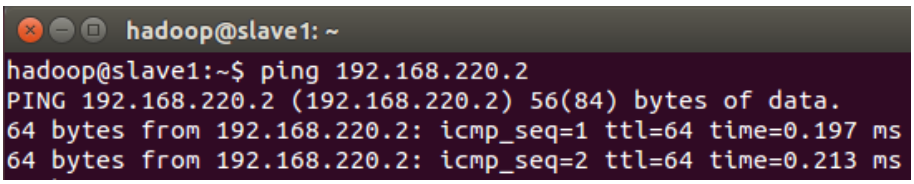


Figura 91 - Ping Slave1 a Master
Fuente: Elaboración Propia

hadoop@ubuntu:~$ ping 192.168.220.3 //Slave1 ping a slave1

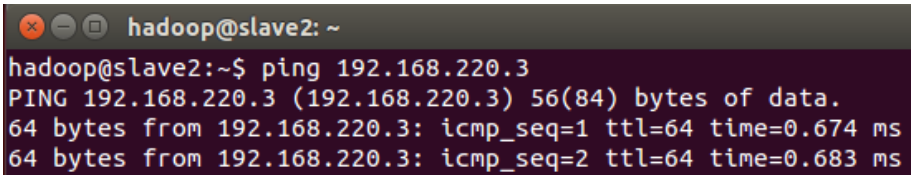


Figura 92 - Ping Slave1 a Slave2
Fuente: Elaboración Propia

Desde el slave2 realizar lo siguiente:

hadoop@ubuntu:~$ ping 192.168.220.2 //Slave2 ping a master

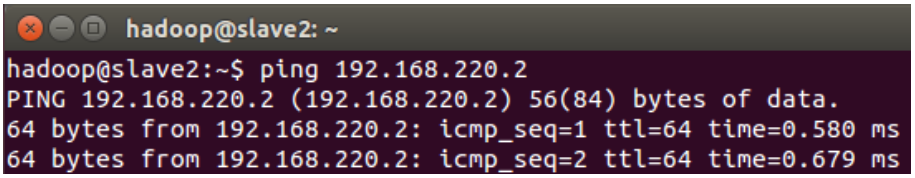


Figura 93 - Ping Slave2 a Master
Fuente: Elaboración Propia

hadoop@ubuntu:~$ ping 192.168.220.4 //Slave2 ping a slave1

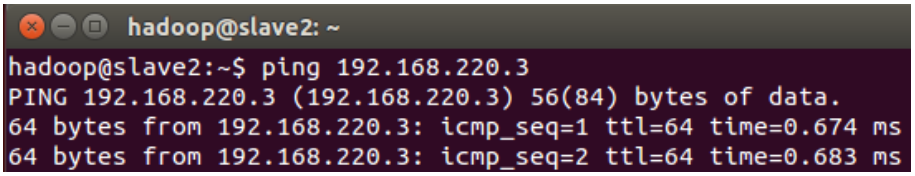


Figura 94 - Ping Slave2 a Slave1
Fuente: Elaboración Propia

3. **Cambiar el nombre de host de los 3 sistemas**

Esto se realiza utilizando el siguiente comando en el terminal:

$ sudo nano /etc/hostname

Escriba 'master' borrando Ubuntu. Presione Ctrl+x en el teclado y luego S para guardar la configuración.

Repita el paso anterior con **slave1** y **slave2** y cambie el nombre de host a slave1, slave2

4. **Actualice los hosts en los 3 nodos.**

Esto se realiza utilizando el siguiente comando en el terminal:

$ sudo vim /etc/hosts

Se abrirá un archivo y se realiza lo siguiente:

127.0.0.1 localhost #no editar esta línea

#127.0.1.1 master #remover esta línea

192.168.220.2 master #añadir esto y las 2 líneas siguientes

192.168.220.3 slave1

192.168.220.4 slave2 #dirección IP y hostname del slave2

Repetir lo anterior en **slave1** y **slave2.**

Reiniciar los equipos para que los cambios hagan efecto.

5. **Confirmar que el hostname de los 3 nodos hayan cambiado.**

   Esto se realiza utilizando el siguiente comando en el terminal:

   $ hostname

   Se debería imprimir master, slave1, slave2 en las 3 máquinas respectivamente.

   De hecho, cuando se ejecuta terminal (ctrl + shift + T), en lugar de mostrar:

   hduser@ubuntu:$

   Se muestra lo siguiente:

   hadoop@master:$ # en el nodo master

   hadoop@slave1:$ # en el nodo slave1

   hadoop@slave2:$ # en el nodo slave2

6. **Realizar ping entre cada uno de los nodos usando el nombre de host.**

   Realizar ping en cada uno de los otros sistemas usando el nombre de host en lugar de la dirección IP.

   Master-> ping hacia el slave1 y slave2

   Slave1-> ping solamente al master

   Slave2-> ping solamente al master

   Esto se realiza utilizando el siguiente comando en el terminal:

   hadoop@master:$ ping slave1

   hadoop@master:$ ping slave2

   hadoop@slave1:$ ping master

   hadoop@slave2:$ ping master

7. **Probar la conectividad SSH.**

   Para probar la conectividad ssh se debe realizar lo siguiente. Al acceder pedirá yes o no y se debe escribir 'yes'. Realice ssh master / slave1 / salve2 en cada uno de los nodos para verificar la conectividad.

   hadoop@master:~$ ssh master

   hadoop@master:~$ ssh slave1 #Escribir yes y se conectará a slave1

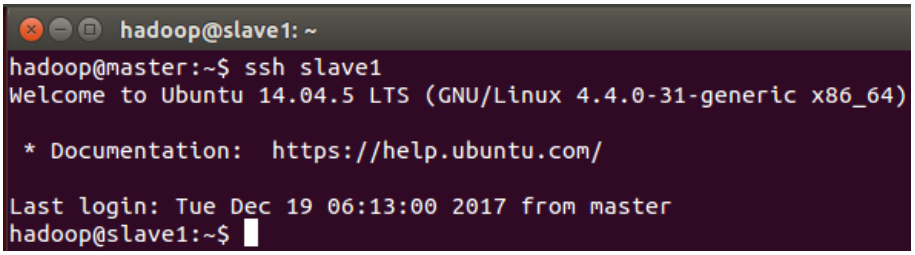


Figura 95 - Conexión ssh Slave1
Fuente: Elaboración Propia

hadoop@slave1:~$ exit #salir de slave1 y volver a master.

hadoop@master:~$ ssh slave2 # Escribir yes y se conectará a slave2

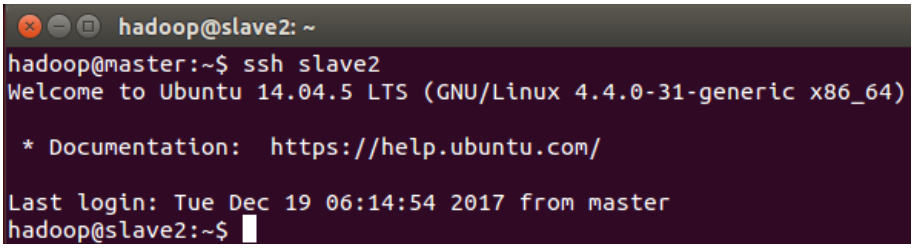


Figura 96 - Conexión ssh Slave2
Fuente: Elaboración Propia

hadoop@slave2:~$ exit # salir de slave1 y volver a master.

hadoop@master:~$

Al ingresar a cada ssh de una de las maquinas pedirá yes o no y se debe escribir 'yes'. Deberíamos ser capaces de ingresar a los ssh master y slaves sin solicitud de contraseña. Si solicita una contraseña mientras se conecta al master o slave usando SSH, hay algo que salió mal y se debe solucionar antes de continuar.

8. **Actualizar core-site.xml (Se debe realizar en el master y en todos los slaves).**

Realizar 2 cambios:

a. Eliminar la configuración de hadoop.tmp.dir. No los necesitamos.
b. Cambiar localhost a maestro.

En el terminal digitar el siguiente comando:

$ sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml

Y editar lo siguiente:

```
~~<property>~~
~~        <name>hadoop.tmp.dir</name>~~
~~        <value>/app/hadoop/tmp</value>~~
~~        <description>A base for other temporary directories.</description>~~
~~</property>~~
 <property>
        <name> fs.default.name </name>
        <value>hdfs://master:9000</value>
```

</property>

El archivo de configuración quedaría de la siguiente manera:

core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
</property>

</configuration>
```

Figura 97 - Configuración core-site.xml multinodo
Fuente: Elaboración Propia

9. **Actualizar hdfs-site.xml (Se debe realizar en el master y en todos los slaves).**

Realizar 3 cambios:

a. La replicación está configurada en 2
b. Namenode configurado solo en el master
c. Datanode configurado solo en slave

En el terminal digitar el siguiente comando:

$ sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Y editar lo siguiente:

```
<property>
   <name>dfs.replication</name>
   <value>2</value> <!--cambiar la replicación de 1 a 2 -->
 </property>
<!--Mantener lo siguiente solamente en el master, y borrar en los slaves-->
<property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<!-- Mantener lo siguiente solamente en los slaves, y borrar en el master -->
```

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
```

El archivo de configuración quedaría de la siguiente manera:

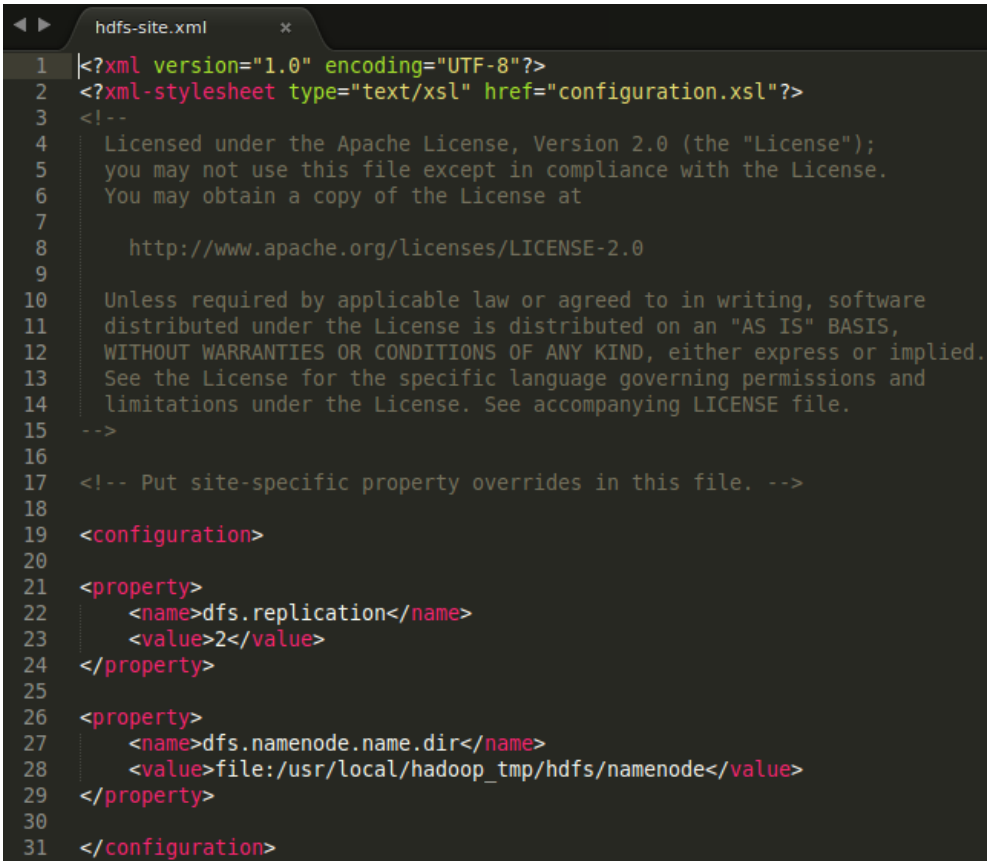


Figura 98 - Configuración hdfs-site.xml multinodo
Fuente: Elaboración Propia

10. **Actualizar yarn-site.xml (Se debe realizar en el master y en todos los slaves).**

    En el terminal digitar el siguiente comando:

    $ sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml

    Y añadir lo siguiente al final de la configuración que se encuentra en el archivo:

```
<property>
        <name>yarn.resourcemanager.resource-
tracker.address</name>
        <value>master:8025</value>
</property>
<property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
</property>
<property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8050</value>
</property>
```

El archivo de configuración quedaría de la siguiente manera:

Figura 99 - Configuración yarn-site.xml multinodo
Fuente: Elaboración Propia

11. **Actualizar mapred-site.xml (Se debe realizar en el master y en todos los slaves).**

En el terminal digitar el siguiente comando:

$ sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml

Modificar localhost por master de la siguiente manera:

```
<property>
        <name>mapreduce.jobhistory.address</name>
        <value>master:10020</value>
</property>
```

El archivo de configuración quedaría de la siguiente manera:

mapred-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>

<property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
</property>

</configuration>
```

Figura 100 - Configuración mapred-site.xml multinodo
Fuente: Elaboración Propia

12. **Actualizar el archivo de masters y slaves (realizar esto solamente en el nodo principal o master)**

Si se encuentra alguna entrada relacionada con localhost hay que eliminarla. Este archivo es solo un archivo auxiliar que utilizan los scripts de hadoop para iniciar los servicios apropiados en los nodos master y slaves.

Ejecutar el siguiente comando en el terminal:

hadoop@master$ sudo nano /usr/local/hadoop/etc/hadoop/slaves

Se abrirá un archive en el cual hay que escribir lo siguiente:

slave1

slave2

Ejecutar el siguiente comando en el terminal:

hadoop@master$ sudo vim /usr/local/hadoop/etc/hadoop/masters

El siguiente archivo no existe por defecto. Así que se creará el archivo. Se abrirá un archive en el cual hay que escribir lo siguiente:

master

Nota: No se necesita configurar lo anterior en los nodos slaves.

13. **Recrear la carpeta de Namenode (realizar solamente en el master).**

    Para realizar esto ejecutar los siguientes comandos en el terminal:

    hadoop@master$ sudo rm -rf /usr/local/hadoop_tmp

    hadoop@master$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode

    hadoop@master$ sudo chown hadoop:hadoop -R /usr/local/hadoop_tmp/

    hadoop@master$ sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode

14. **Recrear la carpeta de Datanode (realizar solamente en los slaves).**

    Para realizar esto ejecutar los siguientes comandos en el terminal:

    hadoop@slave1$ sudo rm -rf /usr/local/hadoop_tmp

    hadoop@slave1$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode

    hadoop@slave1$ sudo chown hadoop:hadoop -R /usr/local/hadoop_tmp/

    hadoop@slave1$ sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode

15. **Formatee el Namenode (realizar esto solamente en el master).**

    Antes de iniciar el clúster, se debe formatear el Namenode usando el siguiente comando solo en el nodo maestro:

    hadoop@master$ hdfs namenode -format

16. **Iniciar el DFS y YARN (realizar esto solamente en el master).**

    hadoop@master$ start-dfs.sh

    hadoop@master$ start-yarn.sh

    En el terminal obtendremos la siguiente respuesta:

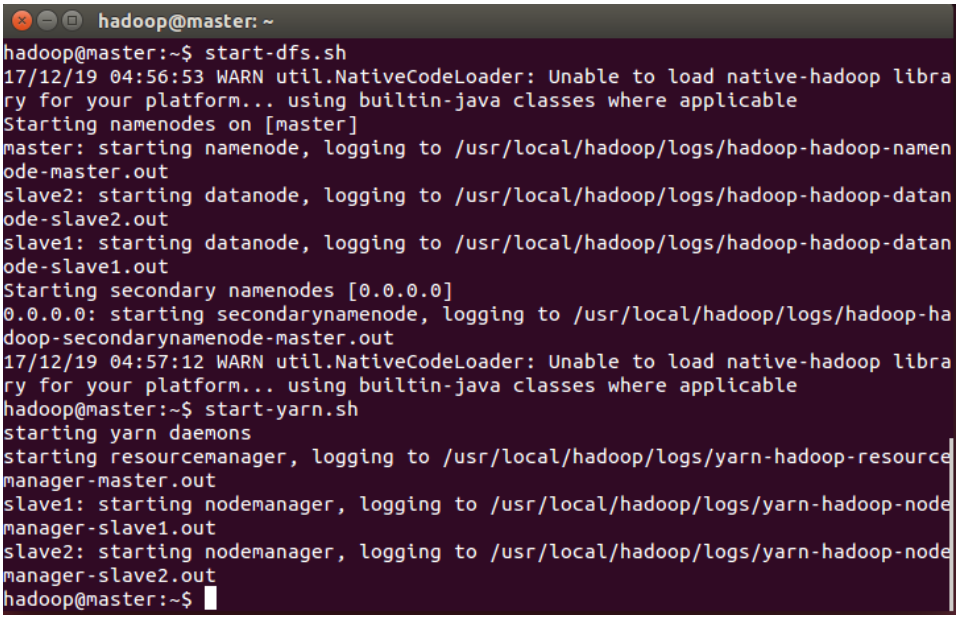


Figura 101 - Levantar Hadoop multinodo
Fuente: Elaboración Propia

Escribir yes cuando se le solicite.

Una vez que se inicia, ejecutar un jps en el master y slaves.

Ejecutar el comando jps en el terminal del master, de la siguiente manera:

hadoop@master$ jps

3379 NameNode

3175 SecondaryNameNode

3539 ResourceManager

Se obtendrá la siguiente respuesta:

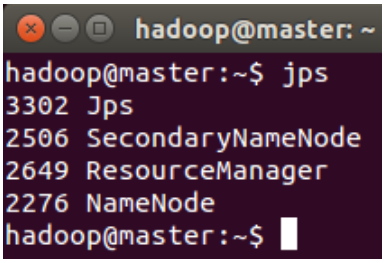


Figura 102 - Servicios levantados Master
Fuente: Elaboración Propia

Ejecutar el comando jps en el terminal del slave1 y slave2, de la siguiente manera:

hadoop@slave1$ jps

2484 DataNode

2607 NodeManager

Se obtendrá la siguiente respuesta:

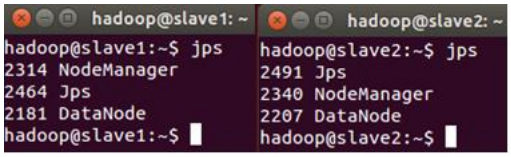


Figura 103 - Servicios levantados Slave1 y Slave2
Fuente: Elaboración Propia

17. **Comprobar si se puede acceder a Hadoop mediante el navegador pulsando las siguientes URL:**

localhost:50070

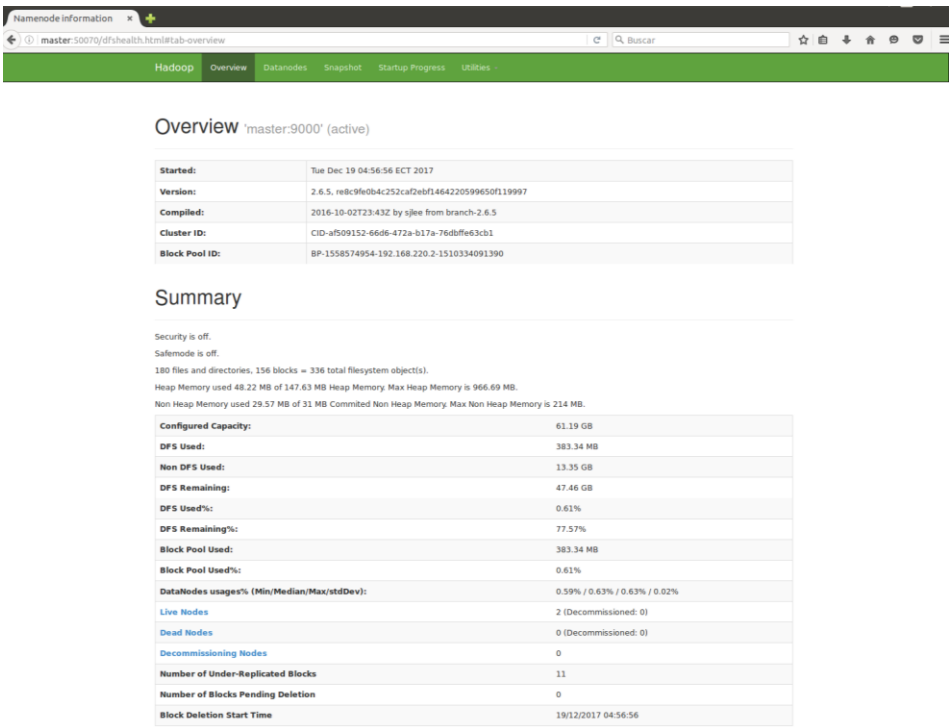


Figura 104 - Comprobar Hadoop Multinodo1
Fuente: Elaboración Propia

La información de los nodos se presenta:

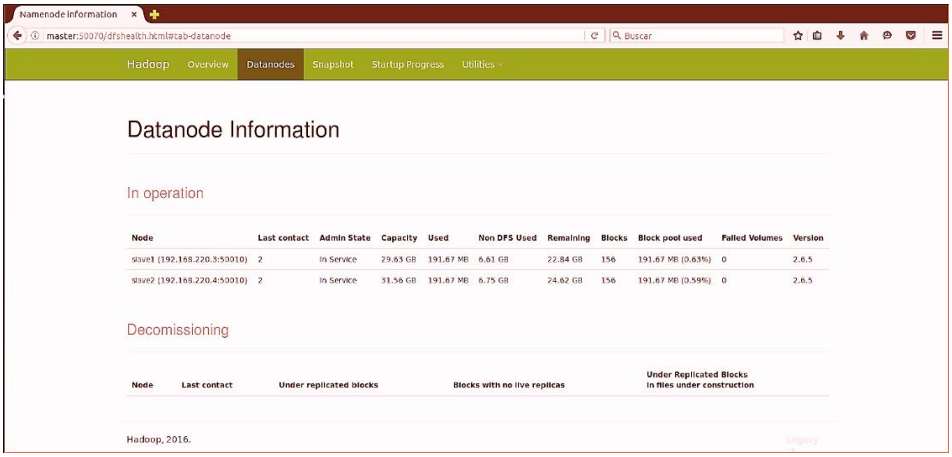


Figura 105 - Comprobar Hadoop Multinodo2
Fuente: Elaboración Propia

localhost:8088

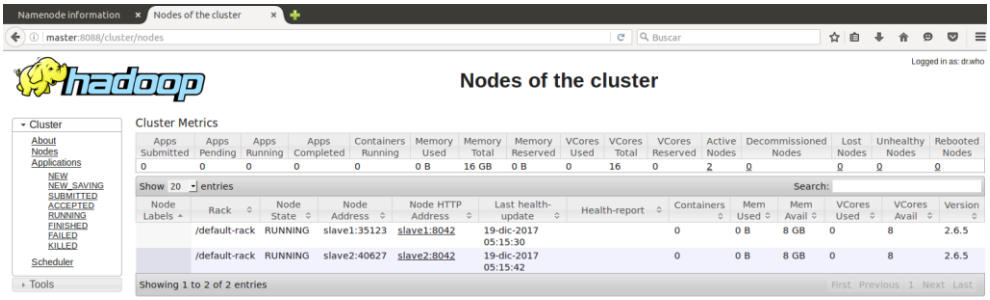


Figura 106 - Comprobar Hadoop Multinodo3
Fuente: Elaboración Propia

Anexo 3: Instalación de R y R Studio

1. Dirigirse a la página oficial de R, mediante el siguiente enlace: https://www.rstudio.com/
2. Buscar la opción de R estudio como en la siguiente imagen y seleccionar download RStudio.

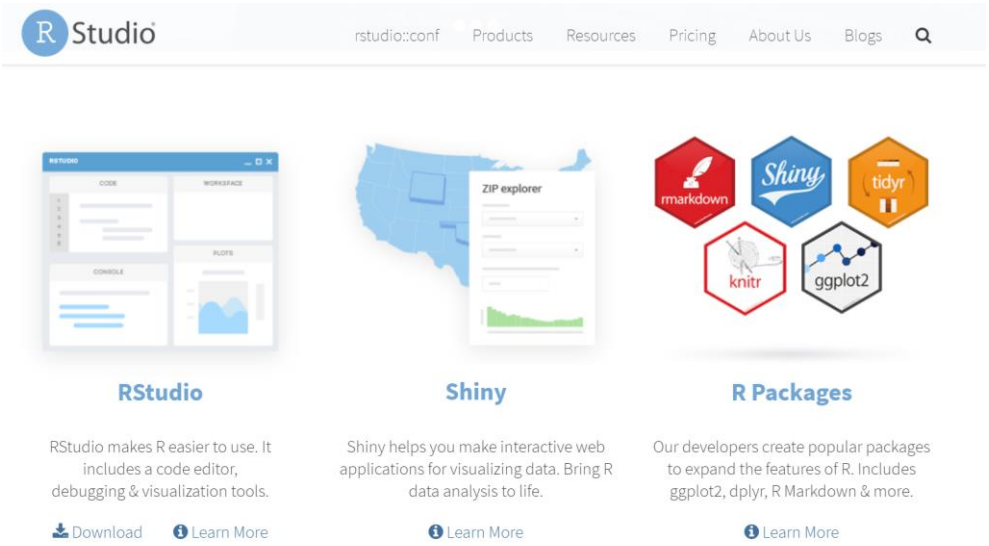
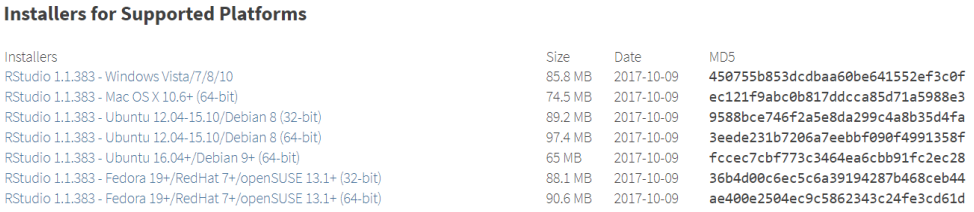


Figura 107 - Sitio de descarga Rstudio
Fuente: Elaboración Propia

3. Seguidamente lo que se debe realizar es descargar un paquete de R que es necesario para el correcto funcionamiento de RStudio. Se lo descarga del siguiente enlace: https://cran.rstudio.com/

Figura 108 - Sitio de Descarga R
Fuente: Elaboración Propia

4. Una vez realizado el paso anterior se procede a la descarga del programa RStudio, la versión que se utiliza es la siguiente:



Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.383 - Windows Vista/7/8/10	85.8 MB	2017-10-09	450755b853dcdbaa60be641552ef3c0f
RStudio 1.1.383 - Mac OS X 10.6+ (64-bit)	74.5 MB	2017-10-09	ec121f9abc0b817ddcca85d71a5988e3
RStudio 1.1.383 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.2 MB	2017-10-09	9588bce746f2a5e8da299c4a8b35d4fa
RStudio 1.1.383 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2017-10-09	3eede231b7206a7eebbf090f4991358f
RStudio 1.1.383 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2017-10-09	fccec7cbf773c3464ea6cbb91fc2ec28
RStudio 1.1.383 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2017-10-09	36b4d00c6ec5c6a39194287b468ceb44
RStudio 1.1.383 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2017-10-09	ae400e2504ec9c5862343c24fe3cd61d

Figura 109 - Descarga RStudio
Fuente: Elaboración Propia

5. Ya que se realizó la descarga de R y RStudio se procede a su instalación haciendo doble clic sobre el instalador descargado.

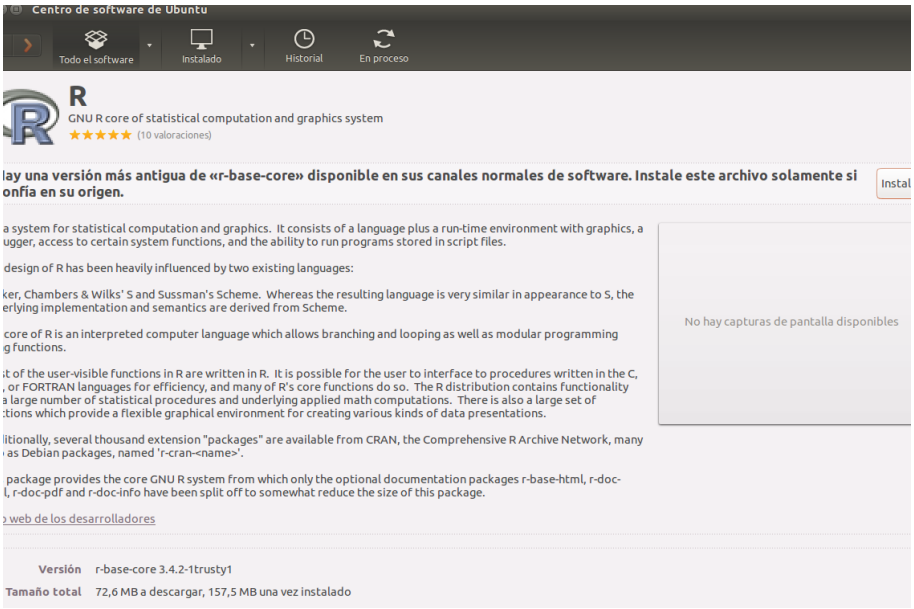


Figura 110 - Instalación R
Fuente: Elaboración Propia

6. Cuando se haya completado la instalación de R se procede a ingresar para constatar su instalación.

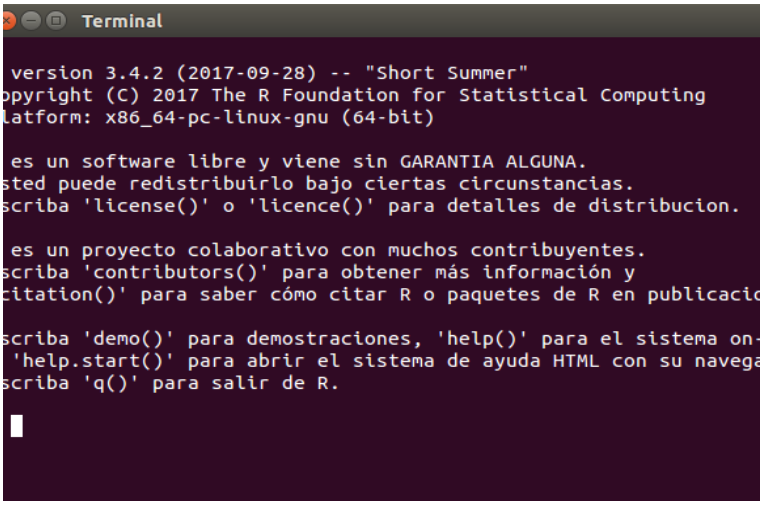


Figura 111 - Interfaz R
Fuente: Elaboración Propia

7. Cuando se haya completado la instalación de RStudio se procede a ingresar para constatar su instalación.

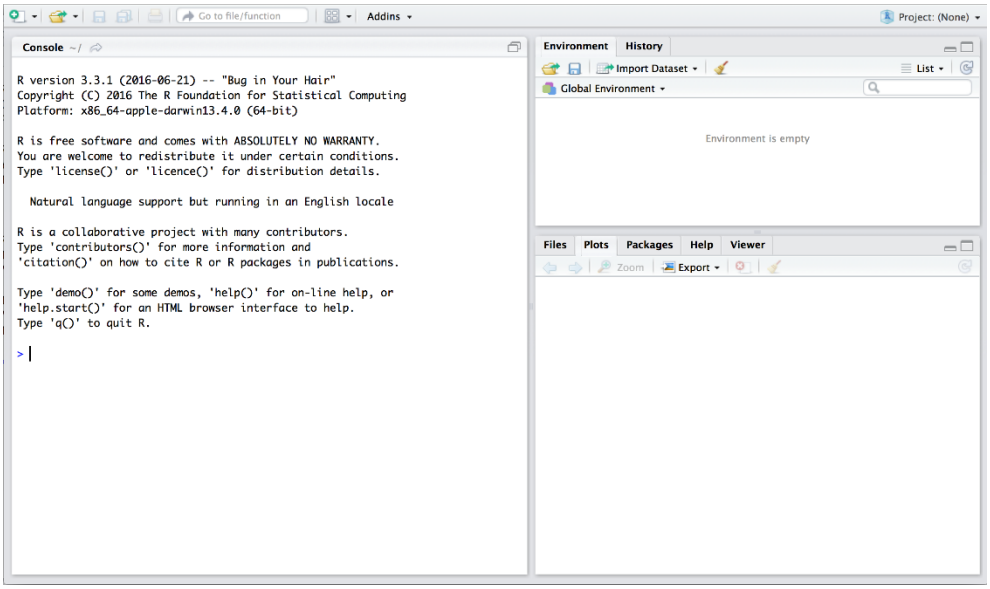


Figura 112 - Interfaz RStudio
Fuente: Elaboración Propia

Anexo 4: Integración entre R y Hadoop.

1. Como primer paso se debe descargar desde el siguiente enlace las librerías rmr y rhdfs que son las necesarias para conseguir la integración entre R y Hadoop. https://github.com/RevolutionAnalytics/RHadoop/wiki

Package Name	Description
rhdfs	This package provides basic connectivity to the Hadoop Distributed File System. R programmers can browse, read, write, and modify files stored in HDFS from within R. Install this package only on the node that will run the R client.
rmr2	A package that allows R developer to perform statistical analysis in R via Hadoop MapReduce functionality on a Hadoop cluster. Install this package on **every node** in the cluster.

Figura 113 - Descarga Librerías RHadoop
Fuente: Elaboración Propia

2. El siguiente paso es configurar java para R, esto se realiza con el siguiente comando desde el terminal de Ubuntu:

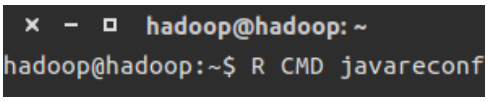


Figura 114 - Configuración Java en R
Fuente: Elaboración Propia

3. En este paso se carga el entorno de Hadoop al archivo de entornos de Ubuntu, esto se realiza mediante el siguiente comando:

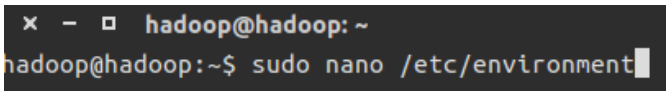


Figura 115 - Abrir archivo de Entornos
Fuente: Elaboración Propia

4. Se añaden las siguientes líneas en el archivo de enviroment, ejecutado en el paso anterior, el archivo debe quedar de la siguiente manera:

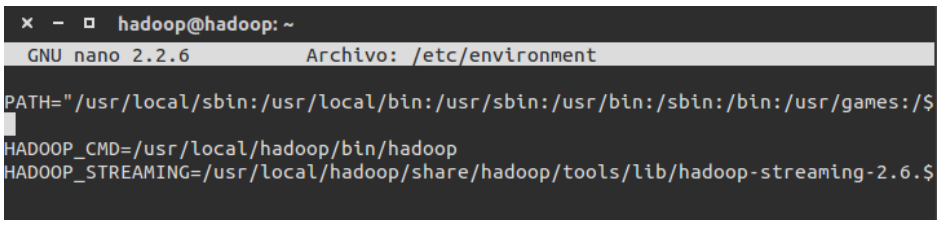


Figura 116 - Añadir entorno de Hadoop
Fuente: Elaboración Propia

5. Luego de haber realizado los pasos anteriores se procede a la instalación de librerías necesarias en la consola de R las cuales sirven para integrar correctamente las librerías de Hadoop antes descargadas, esto se realiza ingresando a la terminal de R y ejecutando las siguientes líneas una por una:

```
> install.packages("reshape2")
> install.packages("Rcpp")
> install.packages("iterators")
> install.packages("itertools")
> install.packages("digest")
> install.packages("RJSONIO")
> install.packages("functional")
```

Figura 117 - Librerías a Instalar en R
Fuente: Elaboración Propia

6. Ya que se realizó la instalación de las librerías antes mencionadas es hora de proceder a instalar las librerías de Hadoop antes descargadas, esto se realiza con la ejecución de las siguientes líneas en la terminal de Ubuntu:

```
hadoop@hadoop: ~
hadoop@hadoop:~$ R CMD INSTALL Descargas/rhdfs_1.0.8.tar.gz
```

Figura 118 - Instalación rhdfs en R
Fuente: Elaboración Propia

```
hadoop@hadoop: ~
hadoop@hadoop:~$ R CMD INSTALL Descargas/rmr2_3.3.1.tar.gz
```

Figura 119 - Instalación rmr en R
Fuente: Elaboración Propia

7. Una vez instaladas las librerías de Hadoop para que funcione integradamente con R se procede a iniciar Hadoop mediante los siguientes comandos:

start-dfs.sh
start-yarn.sh

8. Completado los pasos anteriores es necesario abrir el programa RStudio en el cual se realiza el análisis de los datos.

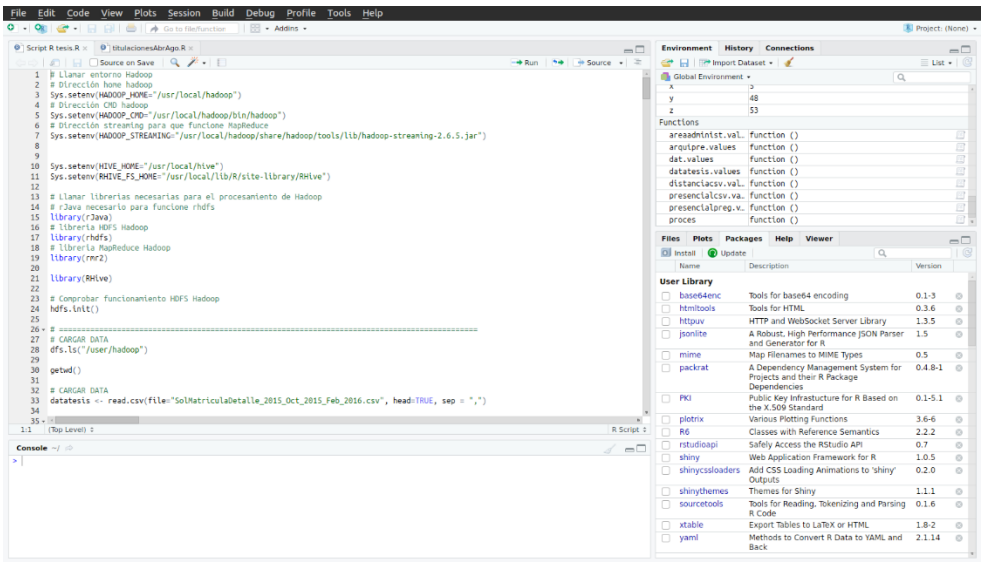


Figura 120 - Ejecutar RStudio
Fuente: Elaboración Propia

9. Una vez dentro del entorno de RStudio se procede a llamar al enviroment o entorno de Hadoop con los siguientes comandos como lo muestra la siguiente imagen:

```
# Llamar entorno Hadoop
# Dirección home hadoop
Sys.setenv(HADOOP_HOME="/usr/local/hadoop")
# Dirección CMD hadoop
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
# Dirección streaming para que funcione MapReduce
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar")
```

Figura 121 - Entorno de Hadoop en RStudio.
Fuente: Elaboración Propia

10. Se debe obtener la siguiente respuesta en la consola de RStudio

```
Console ~/
> Sys.setenv(HADOOP_HOME="/usr/local/hadoop")
> Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
> Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar")
```

Figura 122 - Respuesta Entorno Hadoop
Fuente: Elaboración Propia

11. Comprobar que las librerías rJava rhdfs y rmr2 se encuentren instaladas correctamente, esto se comprueba ejecutando los siguientes comandos como lo muestra la siguiente imagen:

```
# Llamar librerias necesarias para el procesamiento de Hadoop
# rJava necesario para funcione rhdfs
library(rJava)
# libreria HDFS Hadoop
library(rhdfs)
# libreria MapReduce Hadoop
library(rmr2)
```

Figura 123 - Ejecutar Librerías de Hadoop.
Fuente: Elaboración Propia

12. En la consola de Rstudio se deben presentar la ejecución de los comandos sin ningún mensaje de error como lo muestra la siguiente imagen.

```
Console ~/
> library(rJava)
> library(rhdfs)
> library(rmr2)
```

Figura 124 - Respuesta Librerías Hadoop
Fuente: Elaboración Propia

13. Otra forma de comprobar que las librerías se han ejecutado correctamente es buscar en la parte inferior derecha de la interfaz de RStudio, en la pestaña de Paquetes, las librerías rJava, rhdfs y rmr2 deben estar identificadas que se han seleccionado con un check, como lo muestra la siguiente imagen:

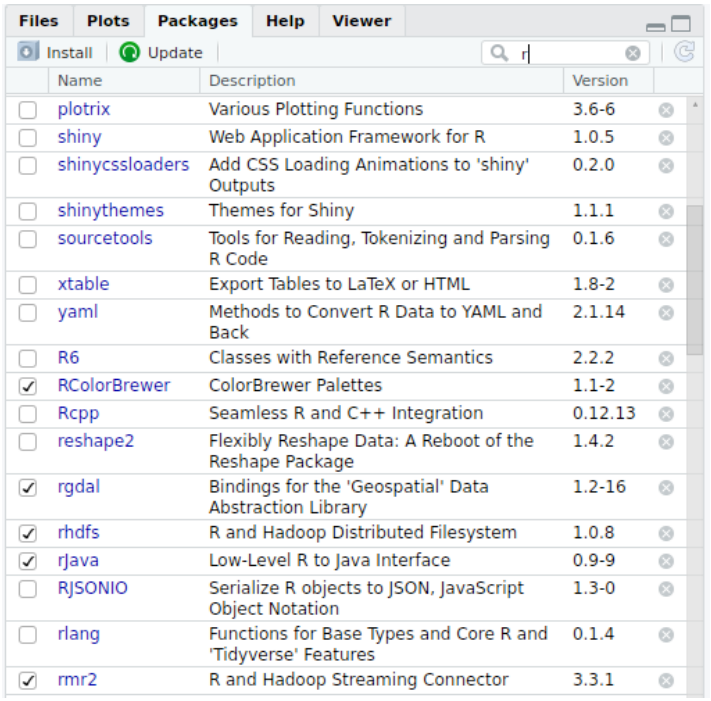


Figura 125 - Pestaña Paquetes Rstudio
Fuente: Elaboración Propia

14. Para realizar la comprobación del funcionamiento de la librería rhdfs de Hadoop se lo realiza con la ejecución del siguiente comando como se indica en la siguiente imagen:

```
# Comprobar funcionamiento HDFS Hadoop
hdfs.init()
```

Figura 126 - Comprobar librería rhdfs de Hadoop
Fuente: Elaboración Propia

15. Se debe obtener la siguiente respuesta en la consola de RStudio sin ningún error:

```
Console ~/
> hdfs.init()
```

Figura 127 - Respuesta RHDFS en consola RStudio
Fuente: Elaboración Propia

16. Para realizar la comprobación del funcionamiento de la librería rmr2 de Hadoop se lo realiza con la ejecución del siguiente comando como se indica en la siguiente imagen:

```
# Comprobar Funcionamiento MapReduce Hadoop
proces <- mapreduce(input=datatesis.values)
```

Figura 128 - Comprobar librería rmr2 de Hadoop
Fuente: Elaboración Propia

17. Se debe obtener la siguiente respuesta en la consola de RStudio sin ningún error:

```
18/01/08 22:46:36 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
packageJobJar: [/tmp/hadoop-unjar5729818492754545878/] [] /tmp/streamjob509864975055388126.jar tmpDir=null
18/01/08 22:46:36 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
18/01/08 22:46:37 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
18/01/08 22:46:38 INFO mapred.FileInputFormat: Total input paths to process : 1
18/01/08 22:46:38 INFO mapreduce.JobSubmitter: number of splits:2
18/01/08 22:46:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1515469491397_0001
18/01/08 22:46:39 INFO impl.YarnClientImpl: Submitted application application_1515469491397_0001
18/01/08 22:46:39 INFO mapreduce.Job: The url to track the job: http://hadoop:8088/proxy/application_1515469491397_0001/
18/01/08 22:46:39 INFO mapreduce.Job: Running job: job_1515469491397_0001
18/01/08 22:46:46 INFO mapreduce.Job: Job job_1515469491397_0001 running in uber mode : false
18/01/08 22:46:46 INFO mapreduce.Job:  map 0% reduce 0%
18/01/08 22:46:57 INFO mapreduce.Job:  map 3% reduce 0%
18/01/08 22:47:06 INFO mapreduce.Job:  map 28% reduce 0%
18/01/08 22:47:09 INFO mapreduce.Job:  map 61% reduce 0%
18/01/08 22:47:12 INFO mapreduce.Job:  map 99% reduce 0%
18/01/08 22:47:13 INFO mapreduce.Job:  map 100% reduce 100%
18/01/08 22:47:13 INFO mapreduce.Job: Job job_1515469491397_0001 completed successfully
18/01/08 22:47:13 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=223630
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=7319320
                HDFS: Number of bytes written=103545511
                HDFS: Number of read operations=14
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=50395
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=50395
                Total vcore-milliseconds taken by all map tasks=50395
                Total megabyte-milliseconds taken by all map tasks=51604480
        Map-Reduce Framework
                Map input records=125
                Map output records=176
                Input split bytes=186
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=182
                CPU time spent (ms)=45290
                Physical memory (bytes) snapshot=498720768
                Virtual memory (bytes) snapshot=2117591040
                Total committed heap usage (bytes)=413663232
        File Input Format Counters
                Bytes Read=7319134
        File Output Format Counters
                Bytes Written=103545511
18/01/08 22:47:13 INFO streaming.StreamJob: Output directory: /tmp/fileebb157ad
```

Figura 129 – Respuesta MapReduce en RStudio
Fuente: Elaboración Propia

Anexo 5: Ejecución de un Script de RHadoop en RStudio.

1. Abrir interfaz de RStudio y proceder a llamar al enviroment de Hadoop de la siguiente manera:

```
Sys.setenv(HADOOP_HOME="/usr/local/hadoop")
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar")
```

Figura 130 - Entorno Hadoop
Fuente: Elaboración Propia

2. Llamar a las librerías de Hadoop que permitirán el procesamiento de los datos, esto se realiza de la siguiente manera:

```
library(rJava)
library(rhdfs)
library(rmr2)
```

Figura 131 - Librerías Hadoop en R
Fuente: Elaboración Propia

3. Ejecutar la siguiente librería, que permite la ejecución de una gráfica en 3D de pastel.

```
library(plotrix)
```

Figura 132 - Librería gráfica 3D
Fuente: Elaboración Propia

4. Se procede a cargar la Data a trabajar y procesar dependiendo del resultado a obtener, esto se realiza de la siguiente manera:

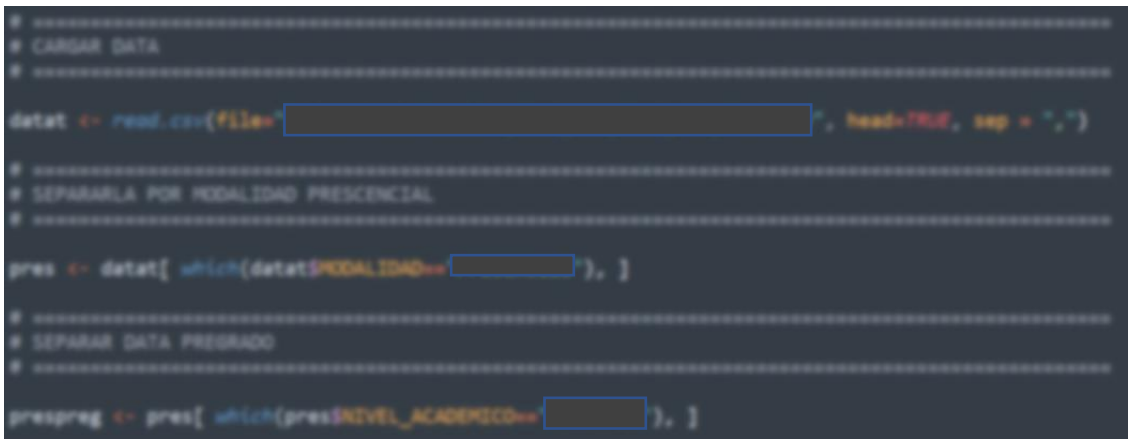


Figura 133 - Cargar Data y Separarla
Fuente: Elaboración Propia

5. Se procede a realizar el procesamiento mediante las librerías de Hadoop, primeramente, se carga el resultado anterior al HDFS, luego se realiza el procesamiento con MapReduce, seguidamente se recupera el procesamiento del HDFS y finalmente el resultado es almaceno en una variable, esto se realiza de la siguiente manera:

```
# Cargar el resultado al HDFS de Hadoop
prespreg.values <- to.dfs(prespreg)
# Realizar el procesamiento con MapReduce de Hadoop
proces <- mapreduce(input=prespreg.values)
# Recuperar el procesamiendo de MapReduce del HDFS de Hadoop
datproc <- from.dfs(proces)
# Cargar el resultado a una variable
presproc <- datproc$val
```

Figura 134 - Procesamiento con Hadoop
Fuente: Elaboración Propia

6. Del resultado del procesamiento con Hadoop realizado anteriormente se procede a realizar el agrupamiento de los datos, esto se realiza de la siguiente manera:

```
# Contador de matriculados por Area
deppre <- table(presproc$      )
# Almacenar el valor mayor a 0 del contador
deppre1 <- deppre[deppre>0]
```

Figura 135 - Filtrado del Resultado
Fuente: Elaboración Propia

7. Luego de realizar el paso anterior, se procede a dividir ese resultado para la media de componentes, en este caso es sobre 6 y se lo redondea para obtener resultados exactos, esto se realiza de la siguiente manera:

```
# Dividir el resultado anterior para la media
matgen <- deppre1/6
# Redondear el valor anterior
matgen <- round(matgen)
```

Figura 136 - Procesar el Resultado
Fuente: Elaboración Propia

8. Una vez obtenido el resultado se procede a generar la gráfica de pasten en 3D de la siguiente manera:

Figura 137 - Proceso para generar gráfica
Fuente: Elaboración Propia

9. Como resultado se obtiene la siguiente imagen en la interfaz de RStudio:

Figura 138 - Gráfica en interfaz de RStudio
Fuente: Elaboración Propia

Anexo 6: Ejecución de un Script de RHadoop desde Prototipo.

1. Para ejecutar un script desde el prototipo realizado en PHP primeramente es necesario ejecutar el servidor donde se encuentra alojado el prototipo, esto se realiza mediante la ejecución del siguiente comando desde la terminal de Linux, se demuestra en la siguiente imagen:

```
hadoop@hadoop:~$ sudo /opt/lampp/lampp start
[sudo] password for hadoop:
```

Figura 139 - Ejecutar Servidor Prototipo
Fuente: Elaboración Propia

2. Es necesario ingresar la contraseña del usuario, si el servidor se inicia normalmente se obtiene la siguiente respuesta en consola:

```
hadoop@hadoop:~$ sudo /opt/lampp/lampp start
[sudo] password for hadoop:
Starting XAMPP for Linux 7.1.11-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...Warning: World-writable config file '/opt/lampp/etc/my
.cnf' is ignored
Warning: World-writable config file '/opt/lampp/etc/my.cnf' is ignored
Warning: World-writable config file '/opt/lampp/etc/my.cnf' is ignored
ok.
```

Figura 140 - Respuesta del Servidor
Fuente: Elaboración Propia

3. Una vez iniciado el servidor se procede ingresar a la siguiente dirección en el navegador la cual aloja el prototipo web. **localhost/tesis/index.php** como respuesta se obtendrá la interfaz principal del prototipo que es la siguiente:

Figura 141 - Interfaz Principal Prototipo
Fuente: Elaboración Propia

4. Cuando se haya ingresado a la interfaz principal se puede seleccionar cualquier tipo de resultado que se desea visualizar, en este ejemplo se elige el resultado general, al seleccionar este tipo de resultado se ingresa a una nueva interfaz que permite seleccionar un tipo de resultado que se desea visualizar, la interfaz de resultados por titulación es la siguiente:

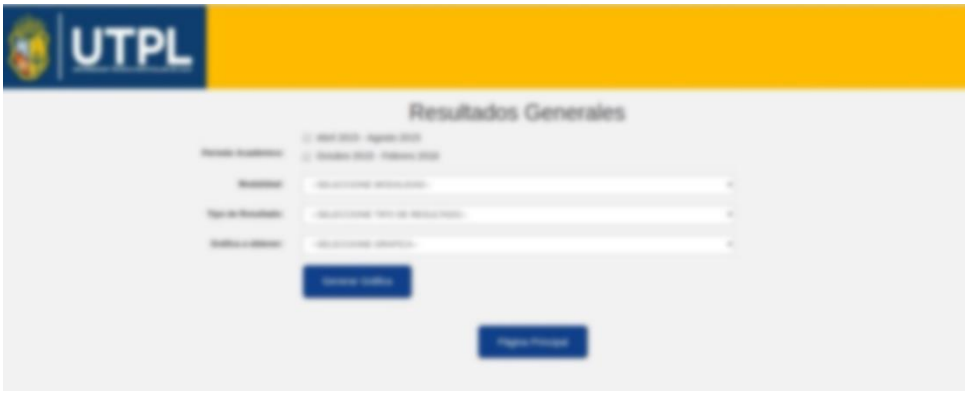


Figura 142 - Interfaz Resultados Generales Prototipo
Fuente: Elaboración Propia

5. Cuando se haya ingresado a la interfaz anterior, se procede a seleccionar el tipo de resultado deseado, la selección debe quedar de la siguiente manera:

Figura 143 - Resultado a Generar
Fuente: Elaboración Propia

6. Cuando se haya seleccionado las opciones como en la imagen anterior se procede a dar click en el botón generar gráfica, se deberá esperar un momento hasta que el script de RHadoop entregue el resultado, el scrip a ejecutarse es el mismo descrito en el Anexo 5, a continuación, se presenta una imagen completa del Script:

Figura 144 - Script a Ejecutar en Prototipo
Fuente: Elaboración Propia

7. Una vez ejecutado el script en segundo plano se obtendrá el siguiente resultado en la interfaz:

Figura 145 - Resultado Script Prototipo
Fuente: Elaboración Propia

Anexo 7: Ejecución de Pruebas de Validación.

7.1. Ejecución de Pruebas

Una vez que se tenga preparada la arquitectura y el ambiente a robar se procede a realizar cada una de las pruebas planteadas, cuyo objetivo principal es descubrir errores que deben ser corregidos para asegurar el correcto funcionamiento de la arquitectura y calidad del sistema multinodo.

7.1.1. Pruebas unitarias

Este tipo de pruebas permitirán comprobar que las funciones del sistema correspondientes a los requerimientos trabajen de manera correcta, recibiendo parámetros y retornando el resultado esperado.

Para iniciar el proceso de las pruebas es necesario que los equipos se encuentren funcionando correctamente y así poder proseguir con cada una de las pruebas unitarias., a continuación, se describe cada una de las pruebas realizadas con el resultado obtenido de las mismas.

7.1.1.1. Prueba unitaria de Comunicación entre equipos de la arquitectura multinodo.

Esta prueba tiene como objetivo comprobar que exista comunicación entre los equipos utilizados de la arquitectura multinodo de Hadoop, para comprobar que existe la correcta comunicación entre los equipos se realiza la ejecución del siguiente comando en la terminal de cada uno de los equipos.

ping "dirección_del_equipo_a_comunicarse" Ejemplo **ping 192.168.220.3**

El resultado de las pruebas de comunicación es la siguiente:

Desde el equipo master realizar lo siguiente:

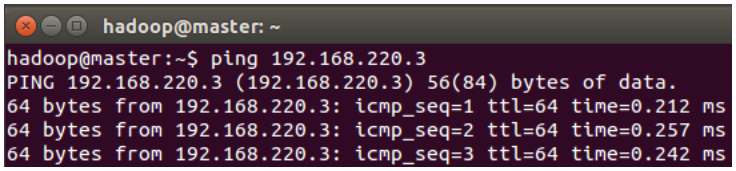


Figura 146 - Prueba ping master a slave1
Fuente: Elaboración Propia

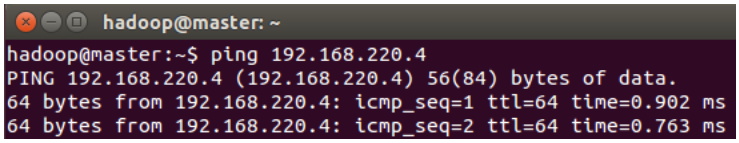


Figura 147 - Prueba ping master a slave2
Fuente: Elaboración Propia

Desde el equipo slave1 realizar lo siguiente:

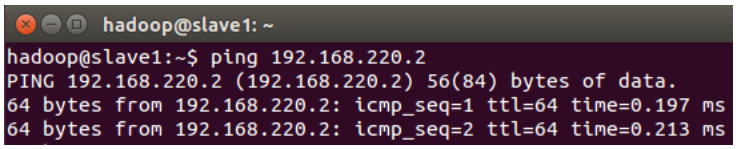


Figura 148 - Prueba ping slave1 a master
Fuente: Elaboración Propia

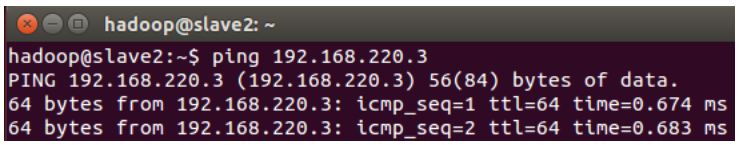


Figura 149 - Prueba ping slave1 a slave2
Fuente: Elaboración Propia

Desde el equipo slave2 realizar lo siguiente:

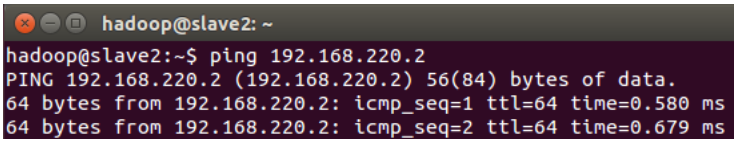


Figura 150 - Prueba ping slave2 a master
Fuente: Elaboración Propia

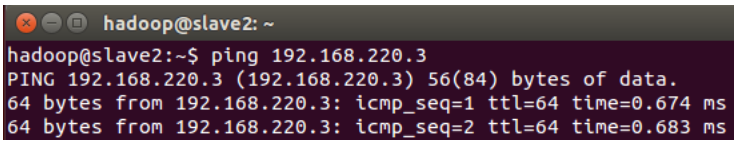


Figura 151 - Prueba ping slave2 a slave1
Fuente: Elaboración Propia

7.1.1.2. Prueba unitaria de Levantar Hadoop multinodo.

Esta prueba tiene como objetivo levantar el servicio de Hadoop multinodo para poder obtener todas las ventajas del funcionamiento de esta arquitectura. Para levantar el servicio de Hadoop se procede a ingresar los siguientes comandos en la terminal del equipo master:

hadoop@master$ start-dfs.sh

hadoop@master$ start-yarn.sh

Como resultado de la ejecución de los comandos anteriormente presentados se debe obtener la siguiente respuesta en consola:

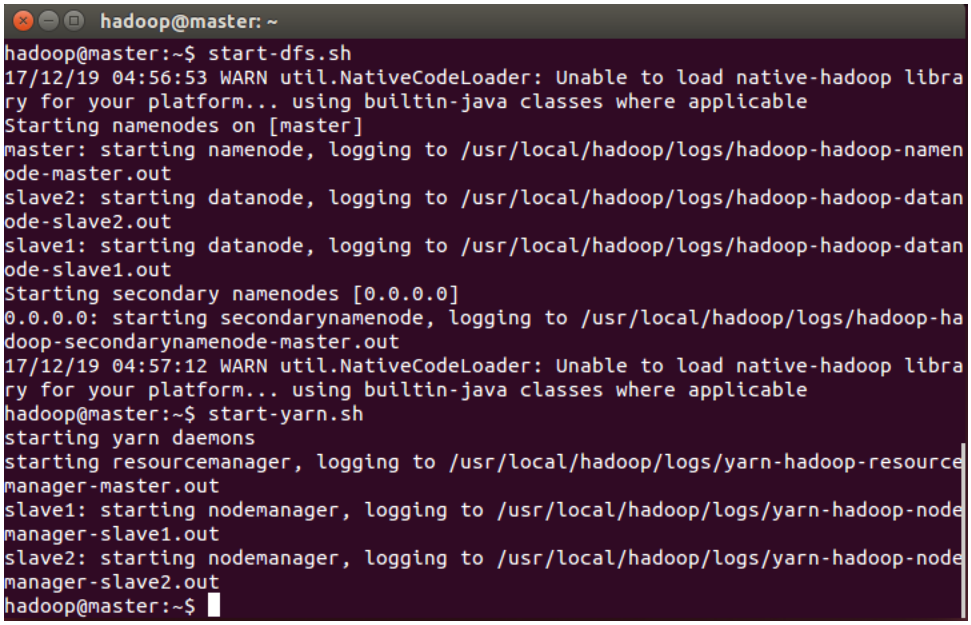


Figura 152 - Levantar Hadoop
Fuente: Elaboración Propia

7.1.1.3. Prueba unitaria de Comprobar en consola servicios Hadoop multinodo levantados.

Esta prueba tiene como objetivo comprobar que los servicios de Hadoop multinodo se hayan levantado correctamente. Para comprobar que los servicios de Hadoop se han levantado correctamente se procede a ingresar el siguiente comando en la terminal de cada uno de los equipos de la arquitectura multinodo:

hadoop@master$ jps

hadoop@slave1$ jps

hadoop@slave2$ jps

Como resultado de la ejecución de los comandos anteriormente presentados se debe obtener la siguiente respuesta en consola de cada uno de los equipos:

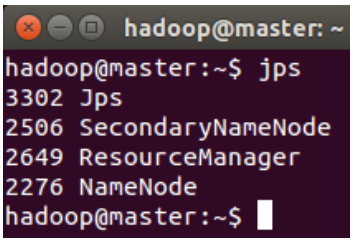


Figura 153 - Servicios master
Fuente: Elaboración Propia

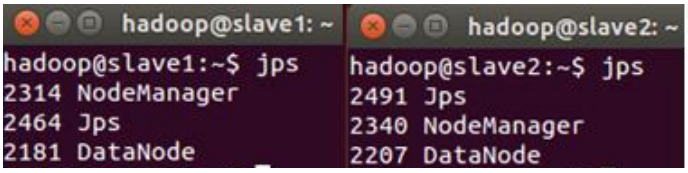


Figura 154 - Servicios Slaves
Fuente: Elaboración Propia

7.1.1.4. Prueba unitaria de Comprobar interfaz de Hadoop.

Esta prueba tiene como objetivo comprobar que Hadoop multinodo se hayan levantado correctamente. Para comprobar que Hadoop se ha levantado correctamente se procede a ingresar a su interfaz con la siguiente dirección:

master:50070

master:8088

Como resultado de la ingresar a la dirección **master:50070** se puede visualizar su interfaz:

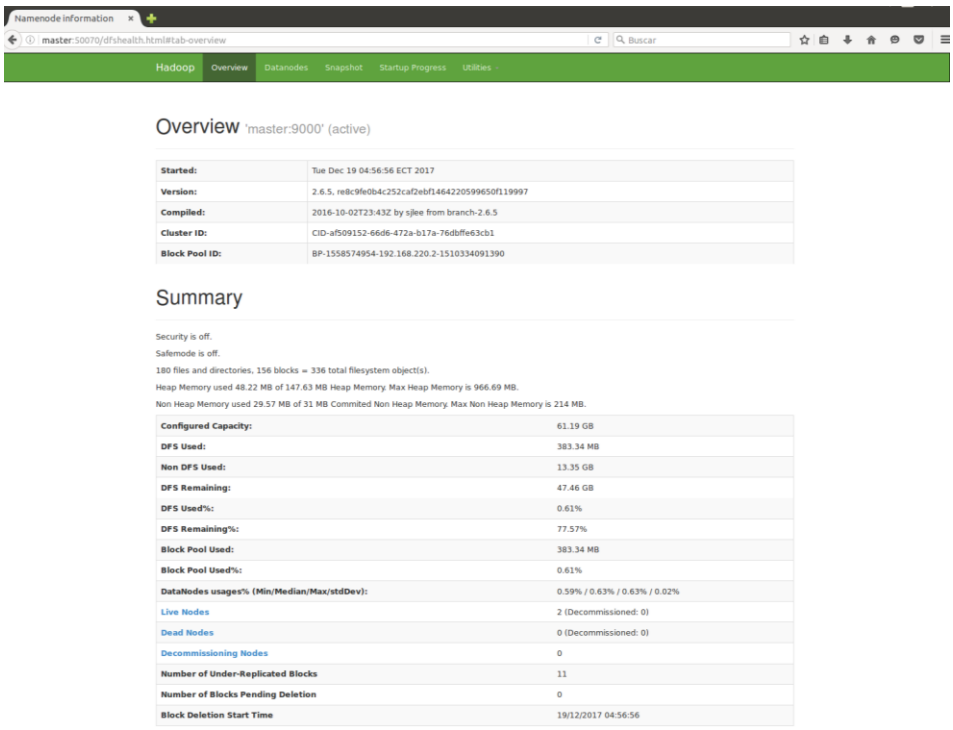


Figura 155 - Informacion Hadoop
Fuente: Elaboración Propia

Como resultado de la ingresar a la dirección **master:8088** se puede visualizar su interfaz:

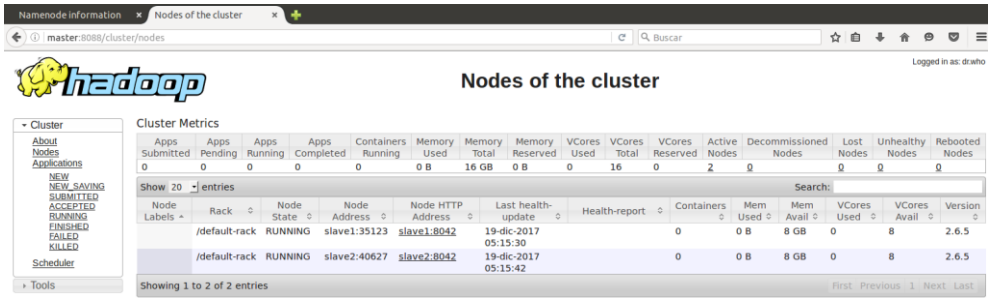


Figura 156 - Aplicaciones Hadoop
Fuente: Elaboración Propia

7.1.1.5. Prueba unitaria de Ejecutar ejemplo funcionamiento de Hadoop multinodo.

Esta prueba tiene como objetivo comprobar que Hadoop multinodo este funcionando correctamente, esta prueba se realiza mediante la ejecución de un ejemplo en la terminal del equipo master, se debe ejecutar la siguiente línea de comando:

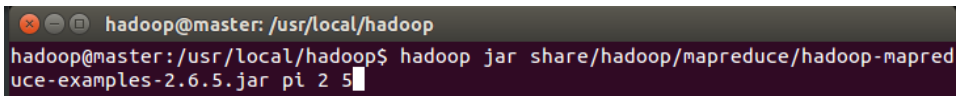


Figura 157 - Ejemplo consola
Fuente: Elaboración Propia

Una vez finalizado el proceso ejecutado se obtiene la siguiente respuesta en el terminal:

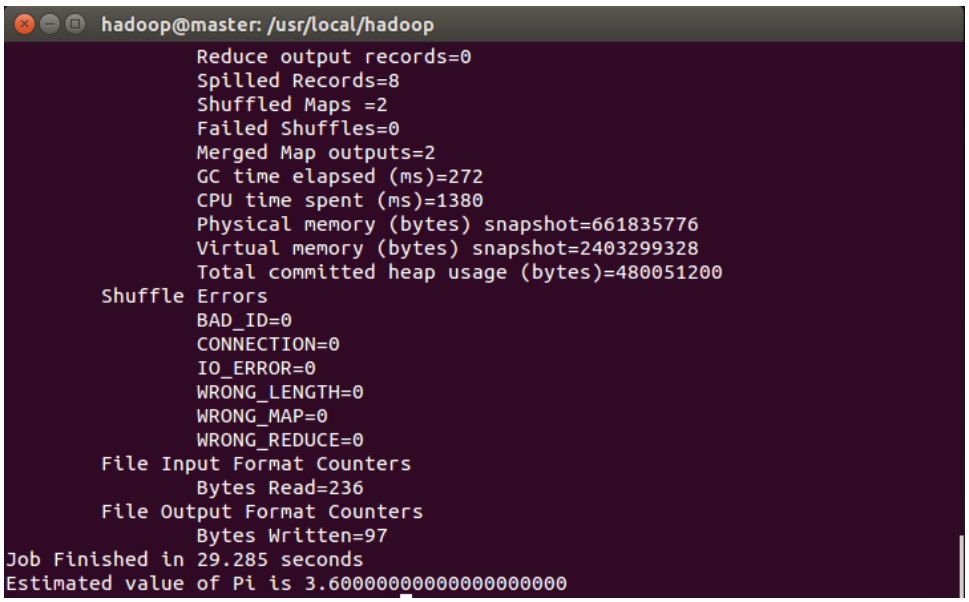


Figura 158 - Resultado Ejemplo Consola
Fuente: Elaboración Propia

7.1.1.6. Prueba unitaria de Comprobar ejecución ejemplo en interfaz de aplicaciones Hadoop.

Esta prueba tiene como objetivo comprobar que Hadoop haya registrado la ejecución del ejemplo realizado en el paso anterior, para eso se debe ingresar a la interfaz de aplicaciones de Hadoop y comprobar que el ejemplo se encuentre registrado, se debe ingresar a la siguiente dirección desde el navegador:

master:8088

El resultado de ingresar a la dirección es el siguiente en donde se comprueba que el ejemplo se encuentra registrado y con una ejecución satisfactoria:

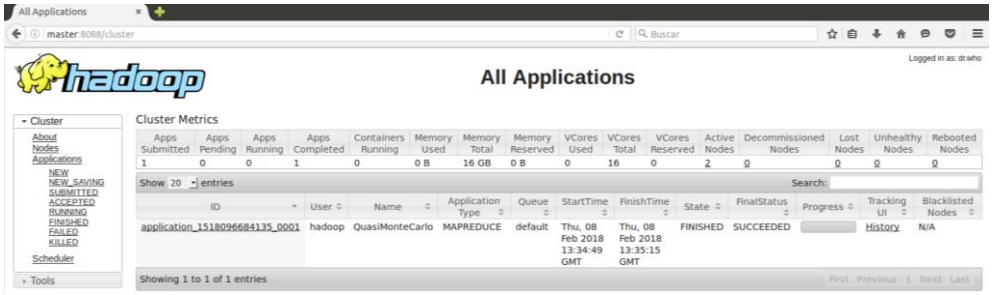


Figura 159 - Registro Ejemplo interfaz
Fuente: Elaboración Propia

7.1.1.7. Prueba unitaria de Comprobar funcionamiento de R.

Esta prueba tiene como objetivo comprobar que el programa R se haya instalado correctamente y los trabajos que se hagan en el brinde los resultados esperados, para cumplir con esta prueba se debe abrir la interfaz en consola de R y se debe ejecutar un ejemplo sencillo de la siguiente manera:

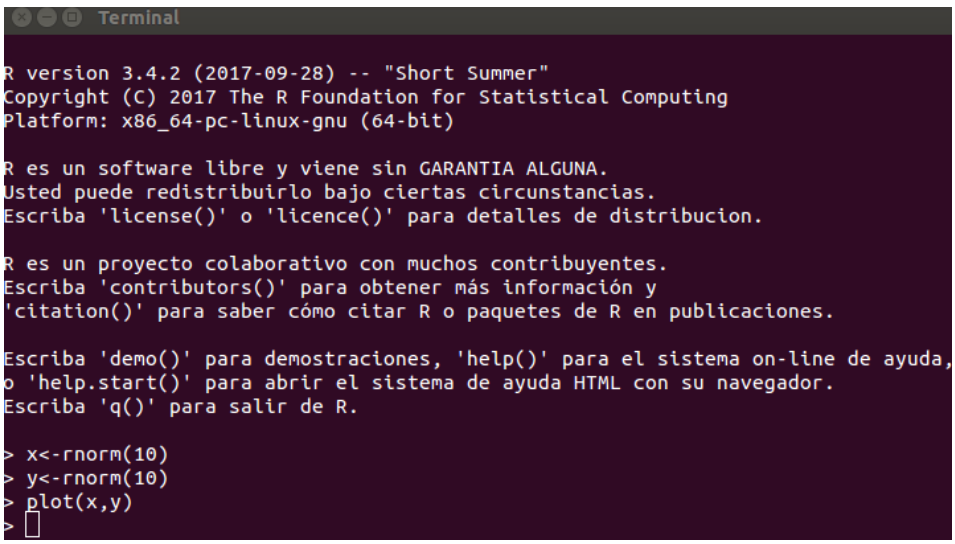


Figura 160 - Ejemplo en R
Fuente: Elaboración Propia

El resultado de la ejecución del ejemplo anterior debe ser el siguiente:

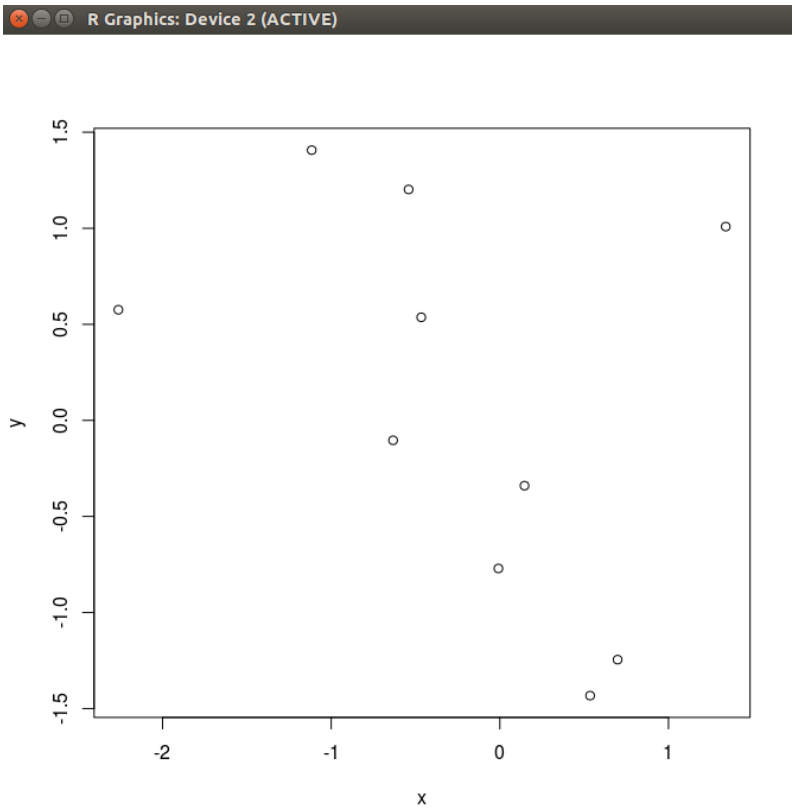


Figura 161 - Resultado Ejemplo R
Fuente: Elaboración Propia

7.1.1.8. Prueba unitaria de Comprobar funcionamiento de RStudio.

Esta prueba tiene como objetivo comprobar que el programa RStudio se haya instalado correctamente y los trabajos que se hagan en el brinde los resultados esperados, para cumplir con esta prueba se debe abrir la interfaz de RStudio y se debe ejecutar un ejemplo sencillo de la siguiente manera:

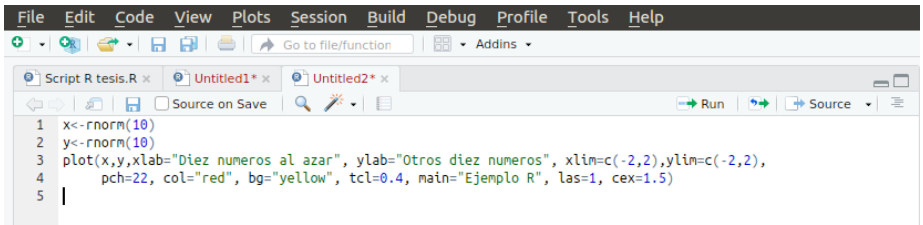


Figura 162 - Ejemplo RStudio
Fuente: Elaboración Propia

El resultado en la interfaz de RStudio de la ejecución del ejemplo anterior debe ser el siguiente:

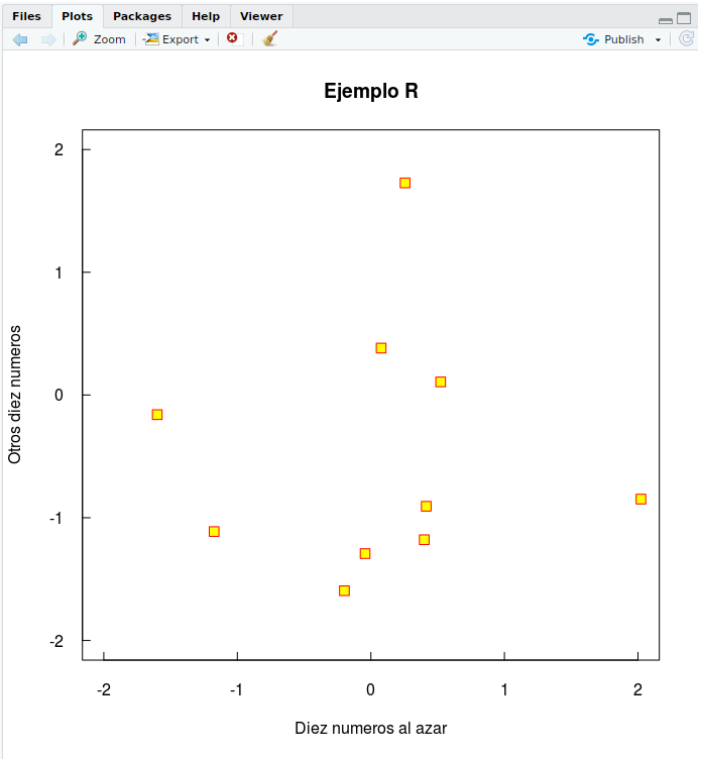


Figura 163 - Resultado Ejemplo RStudio
Fuente: Elaboración Propia

7.1.1.9. Prueba unitaria de Levantar entorno de Hadoop en RStudio.

Esta prueba tiene como objetivo comprobar que el entorno de Hadoop se levante correctamente sin ningún problema en el programa RStudio, para cumplir con esta prueba se debe abrir la interfaz de RStudio y se debe ejecutar las siguientes líneas de comandos:

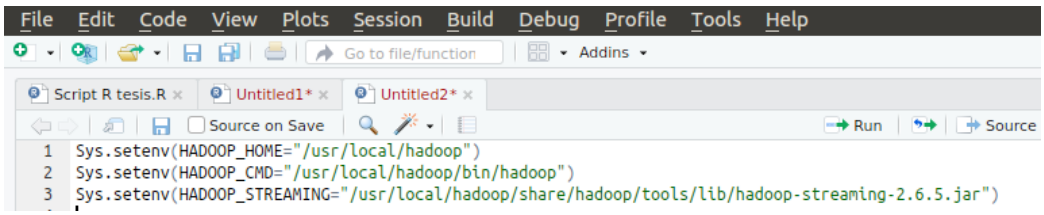


Figura 164 - Entorno RHadoop
Fuente: Elaboración Propia

El resultado de la ejecución de las líneas de comandos debe dar como resultado una respuesta sin ningún error en la consola de RStudio de la siguiente manera:

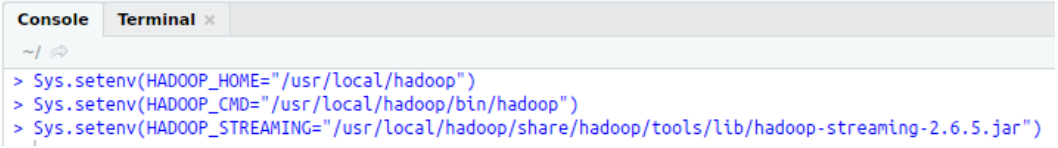


Figura 165 - Resultado Entorno RHadoop
Fuente: Elaboración Propia

7.1.1.10. Prueba unitaria de Comprobar funcionamiento de las librerías de RHadoop en RStudio.

Esta prueba tiene como objetivo comprobar que las librerías de Hadoop se levanten correctamente sin ningún problema en el programa RStudio, así se cumple con el fin de integración entre R y Hadoop, para cumplir con esta prueba se debe abrir la interfaz de RStudio y se debe ejecutar las siguientes líneas de comandos:

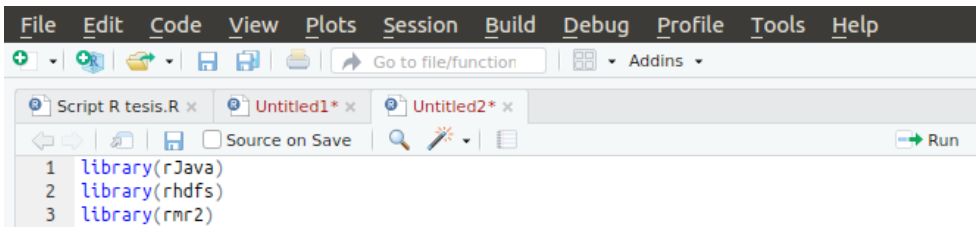


Figura 166 - Librerías RHadoop
Fuente: Elaboración Propia

El resultado de la ejecución de las líneas de comandos debe dar como resultado una respuesta sin ningún error en la consola de RStudio de la siguiente manera:

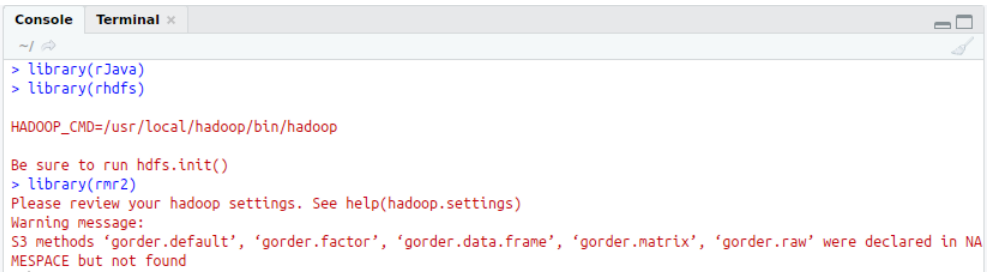


Figura 167 - Resultado Librerías RHadoop
Fuente: Elaboración Propia

7.1.1.11. Prueba unitaria de Comprobar que la Data a trabajar se cargue correctamente en RStudio.

Esta prueba tiene como objetivo comprobar que la data a trabajar se pueda cargar exitosamente, esta prueba se realiza mediante la ejecución de la siguiente línea de comando en la interfaz de RStudio:

Figura 168 - Lectura Data
Fuente: Elaboración Propia

El resultado de la ejecución del comando anterior debe ser satisfactorio y para comprobar que la data este cargada correctamente se procede a visualizarla de la siguiente manera:

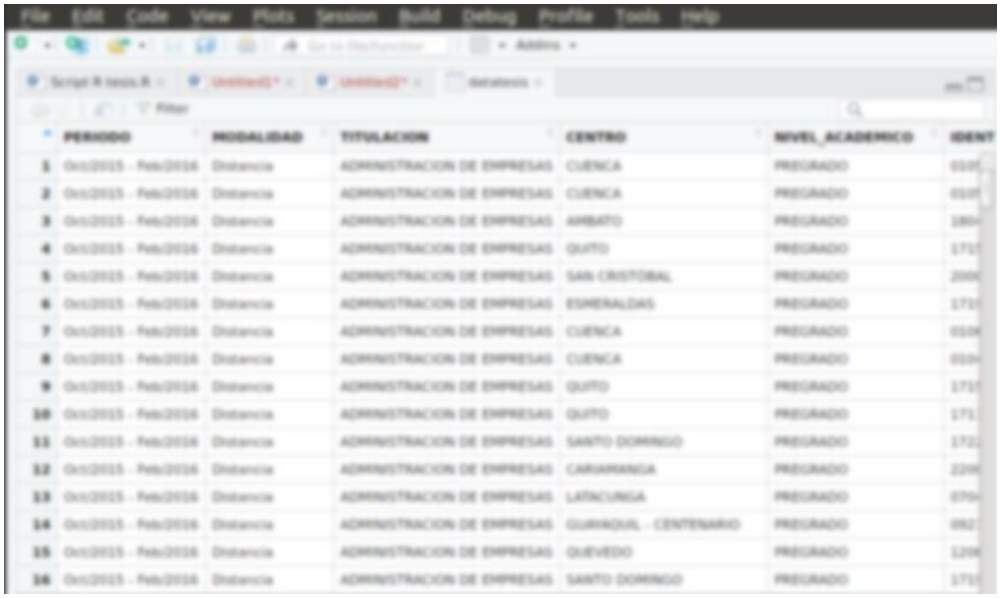


Figura 169 - Visualización Data
Fuente: Elaboración Propia

7.1.1.12. Prueba unitaria de Comprobar procesamiento de la Data con librerías de RHadoop en RStudio.

Esta prueba tiene como objetivo comprobar que las librerías encargadas de la integración y funcionalidad de RHadoop estén correctamente integradas y sean funcionales para el multiprocesamiento de la data, esta prueba se realiza mediante la ejecución de las librerias en la interfaz de RStudio de la siguiente manera:

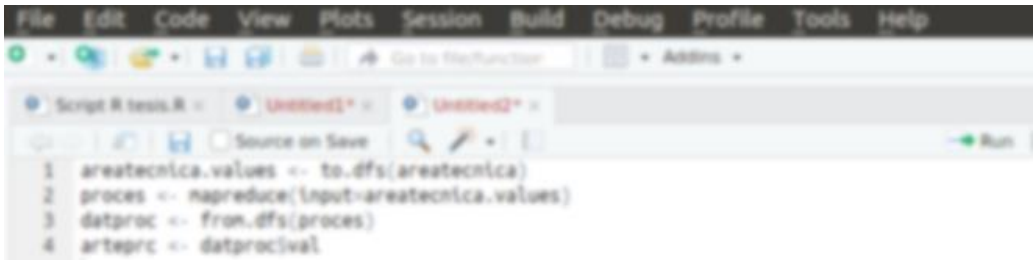


Figura 170 - Procesamiento RHadoop
Fuente: Elaboración Propia

El resultado de la ejecución de las librerías de RHadoop explicadas anteriormente deben presentar un mensaje satisfactorio y no deben presentar ningún error como en la siguiente imagen:

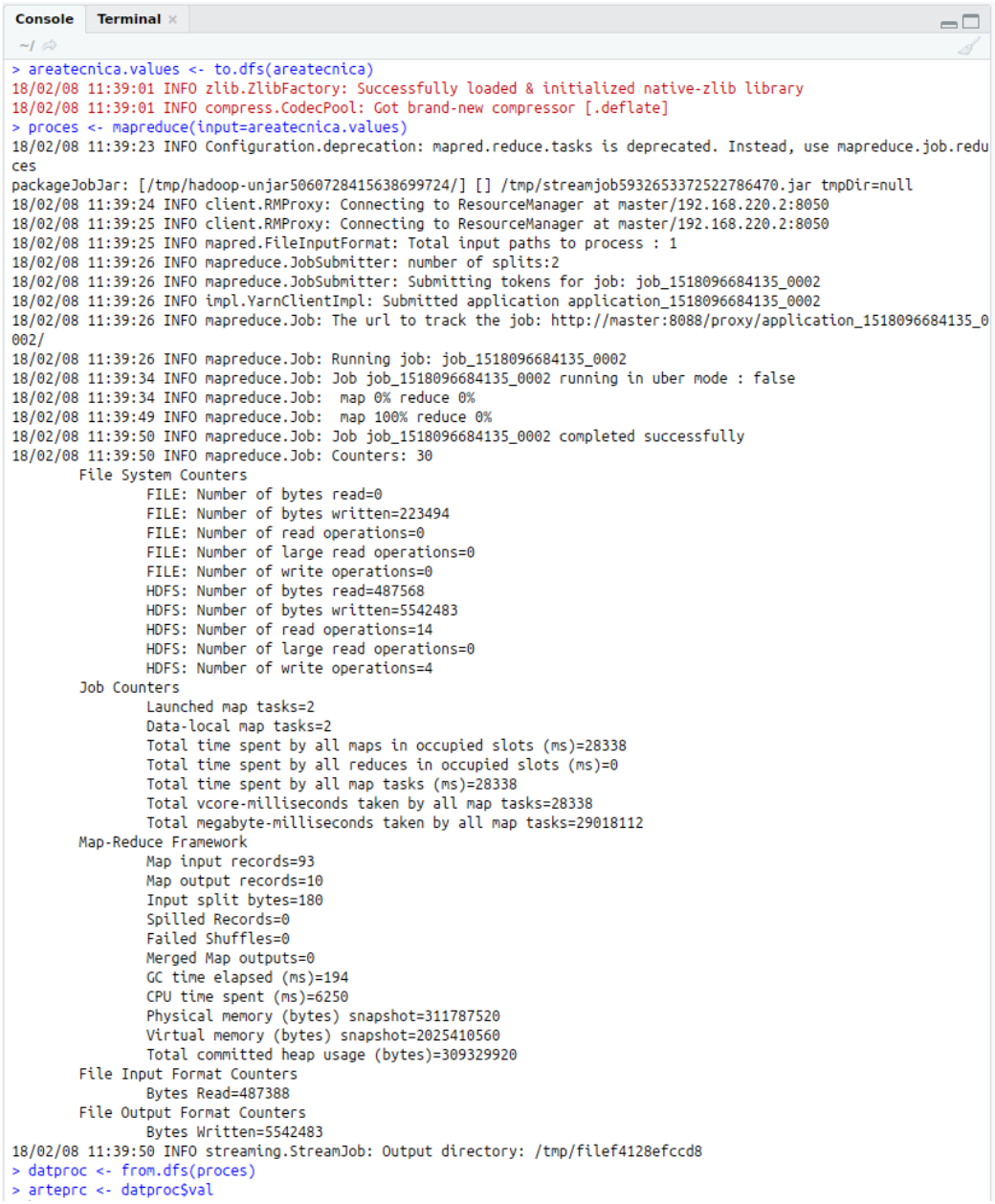


Figura 171 - Resultado Procesamiento RHadoop
Fuente: Elaboración Propia

7.1.1.13. Prueba unitaria de Recuperar el procesamiento de la data y generar una gráfica.

Esta prueba tiene como objetivo comprobar que con el resultado obtenido del procesamiento de las librerías de RHadoop es posible trabajar con ese resultado y poder generar una gráfica, se deben ejecutar las siguientes líneas para procesar ese resultado y generar una gráfica:

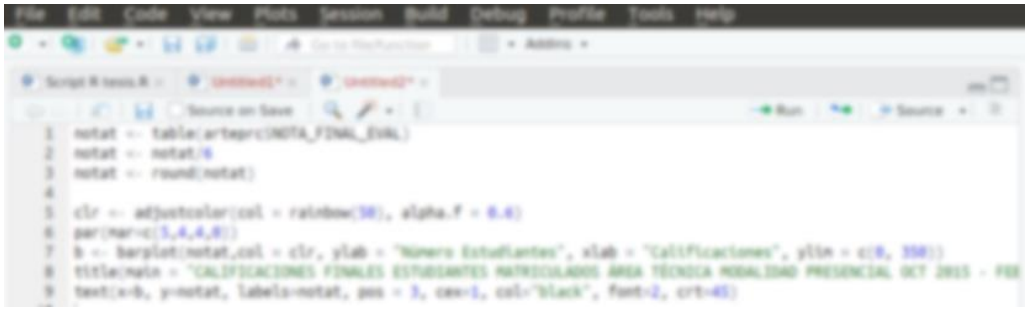


Figura 172 - Filtrado Data
Fuente: Elaboración Propia

Al cumplir con la ejecución de las líneas anteriores se presenta el siguiente resultado:

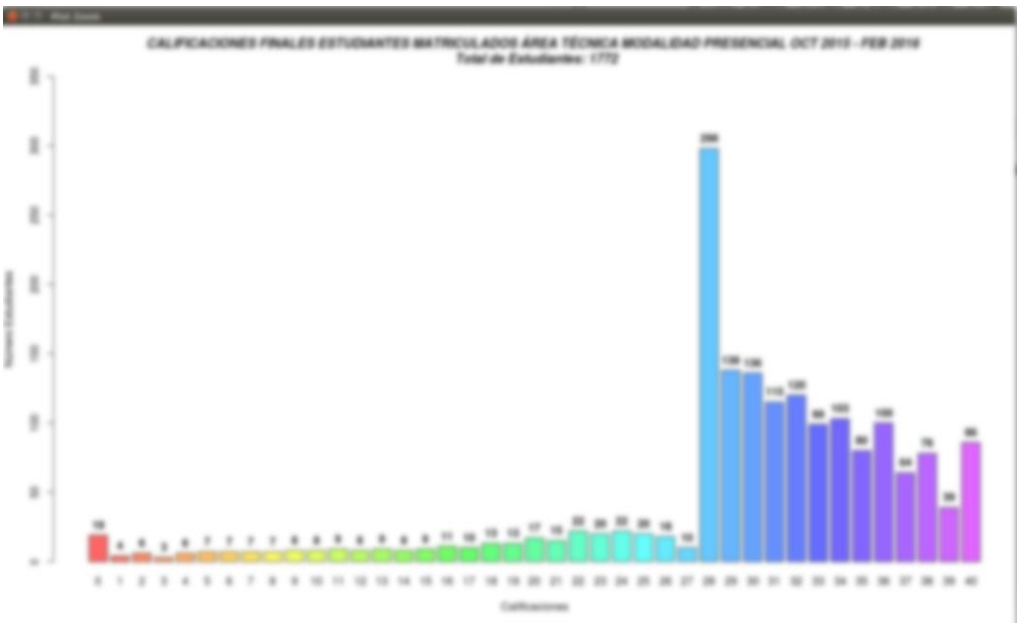


Figura 173 - Resultado Gráfica
Fuente: Elaboración Propia

7.1.2. Pruebas de Sistema.

Las pruebas de sistema permiten comprobar que la arquitectura de Hadoop funcione de manera correcta, y que el resultado del procesamiento de la herramienta RHadoop devuelva los resultados esperados y generen las gráficas correctamente. El ambiente de las pruebas de sistema se ha realizado en arquitecturas con máquinas virtuales y una con sistema operativo nativo con el sistema operativo Ubuntu 14.04, las pruebas se realizan con la arquitectura Hadoop nodo singular versus Hadoop multinodo, permitiendo comprobar el tiempo de respuesta de cada una de las arquitecturas.

Se ha realizado una serie de 50 pruebas por cada uno de los resultados presentados a continuación, los cuales indican el tiempo de procesamiento de la data con el algoritmo de MapReduce de Hadoop, se ha seleccionado una prueba de cada una de las realizadas para ser presentadas a continuación.

7.1.2.1. Prueba data completa 5 gigabytes.

La siguiente prueba indica el tiempo del procesamiento de 5Gb de la data por MapReduce, se puede visualizar el tiempo que el equipo ha utilizado para lograr el correcto procesamiento de la misma, el tiempo se encuentra especificado por la etiqueta CPU time spend (tiempo gastado por el CPU).

```
Map-Reduce Framework
        Map input records=125
        Map output records=176
        Input split bytes=188
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=678
        CPU time spent (ms)=75760
```

Figura 174 - Proceso Multinodo Máquina Virtual 5Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=125
        Map output records=176
        Input split bytes=180
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=945
        CPU time spent (ms)=108300
```

Figura 175 - Proceso Nodo Singular Máquina Virtual 5Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=125
        Map output records=176
        Input split bytes=188
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=478
        CPU time spent (ms)=55760
```

Figura 176 - Proceso Nodo Singular Nativo 5Gb
Fuente: Elaboración Propia

Los resultados del procesamiento anterior se presentan en la siguiente gráfica estadística:

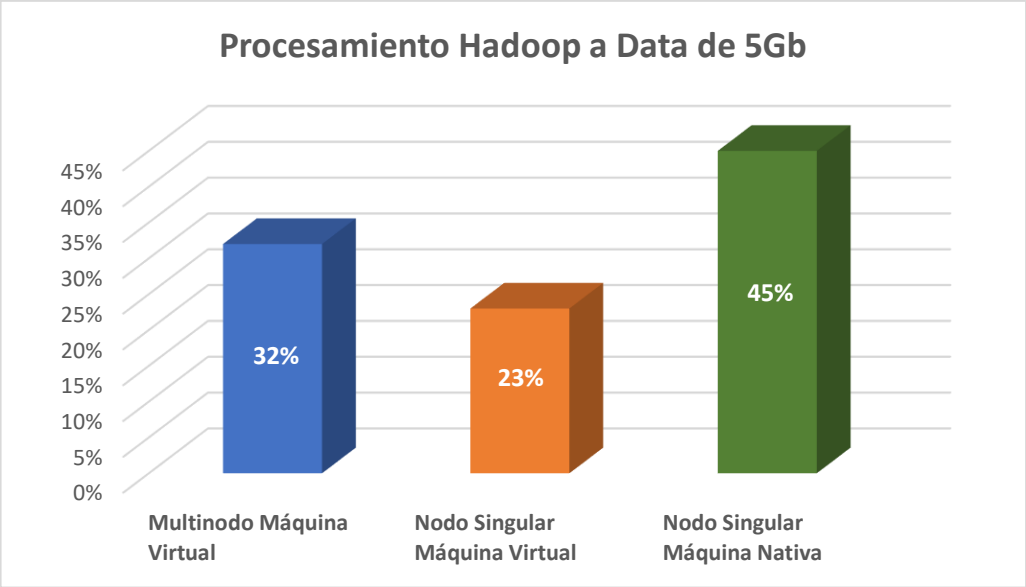


Figura 177 - Resultado Procesamiento 5Gb
Fuente: Elaboración Propia

Los resultados presentados en la figura 177, se pueden interpretar que el procesamiento de la data es mucho más veloz en una arquitectura de nodo singular en una máquina nativa, ya que el equipo tiene mayor poder de procesamiento y velocidad que una máquina virtual, pero en comparación a la prueba en equipos virtuales se demuestra que el procesamiento es mucho más veloz en una arquitectura multinodo en comparación a una de nodo singular.

La máquina nativa realiza el procesamiento de 5Gb de datos en un promedio de 55 segundos a un minuto 20 segundos, mientras que la arquitectura multinodo en máquinas virtuales lo realiza en un promedio de un minuto 20 a dos minutos y la arquitectura de nodo singular en máquina virtual lo realiza en un promedio de dos a tres minutos.

7.1.2.2. Prueba data filtrada 3 gigabytes.

La siguiente prueba indica el tiempo del procesamiento de 3Gb de la data por MapReduce, se puede visualizar el tiempo que el equipo ha utilizado para lograr el correcto procesamiento de la misma, el tiempo se encuentra especificado por la etiqueta CPU time spend (tiempo gastado por el CPU).

```
Map-Reduce Framework
        Map input records=135
        Map output records=146
        Input split bytes=180
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=657
        CPU time spent (ms)=68130
```

Figura 178 - Proceso Multinodo Máquina Virtual 3Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=135
        Map output records=146
        Input split bytes=188
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=807
        CPU time spent (ms)=96750
```

Figura 179 - Proceso Nodo Singular Máquina Virtual 3Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=135
        Map output records=146
        Input split bytes=180
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=475
        CPU time spent (ms)=41380
```

Figura 180 - Proceso Nodo Singular Nativo 3Gb
Fuente: Elaboración Propia

Los resultados del procesamiento anterior se presentan en la siguiente gráfica estadística:

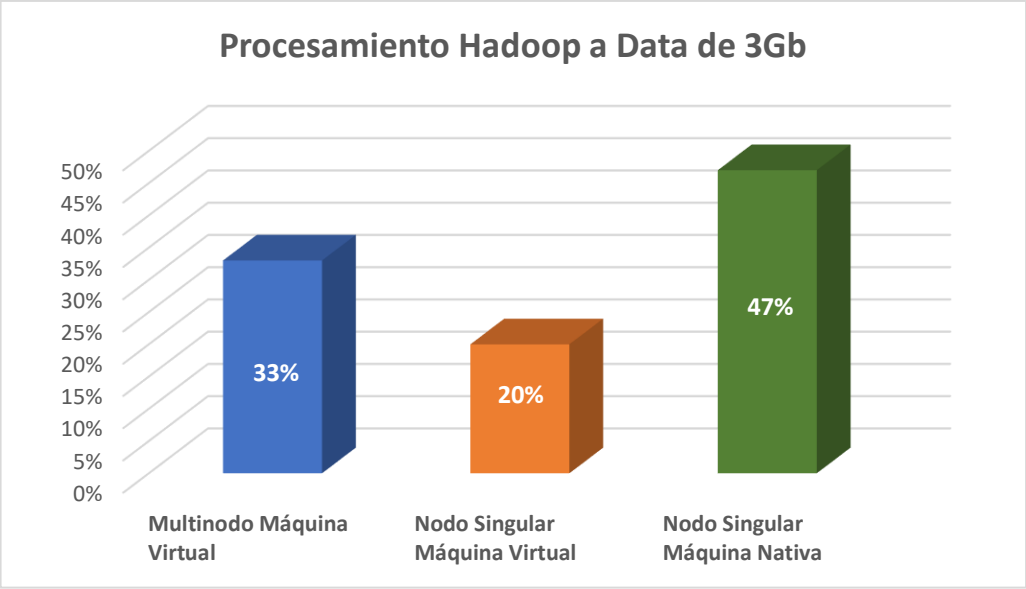


Figura 181 - Resultado Procesamiento 3Gb
Fuente: Elaboración Propia

Los resultados presentados en la figura 181, se pueden interpretar que el procesamiento de la data es mucho más veloz en una arquitectura de nodo singular en una máquina nativa, ya que el equipo tiene mayor poder de procesamiento y velocidad que una máquina virtual, pero en comparación a la prueba en equipos virtuales se demuestra que el procesamiento es mucho más veloz en una arquitectura multinodo en comparación a una de nodo singular.

La máquina nativa realiza el procesamiento de 3Gb de datos en un promedio de 42 segundos a un minuto, mientras que la arquitectura multinodo en máquinas virtuales lo realiza en un promedio de un minuto 3 segundos a un minuto y medio, y la arquitectura de nodo singular en máquina virtual lo realiza en un promedio de minuto y medio a dos minutos.

7.1.2.3. Prueba data filtrada 1 gigabytes.

La siguiente prueba indica el tiempo del procesamiento de 1Gb de la data por MapReduce, se puede visualizar el tiempo que el equipo ha utilizado para lograr el correcto procesamiento de la misma, el tiempo se encuentra especificado por la etiqueta CPU time spend (tiempo gastado por el CPU).

```
Map-Reduce Framework
        Map input records=99
        Map output records=24
        Input split bytes=180
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=546
        CPU time spent (ms)=55806
```

Figura 182 - Proceso Multinodo Máquina Virtual 1Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=99
        Map output records=24
        Input split bytes=188
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=749
        CPU time spent (ms)=82409
```

Figura 183 - Proceso Nodo Singular Máquina Virtual 1Gb
Fuente: Elaboración Propia

```
Map-Reduce Framework
        Map input records=99
        Map output records=24
        Input split bytes=180
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=346
        CPU time spent (ms)=31806
```

Figura 184 - Proceso Nodo Singular Nativo 1Gb
Fuente: Elaboración Propia

Los resultados del procesamiento anterior se presentan en la siguiente gráfica estadística:

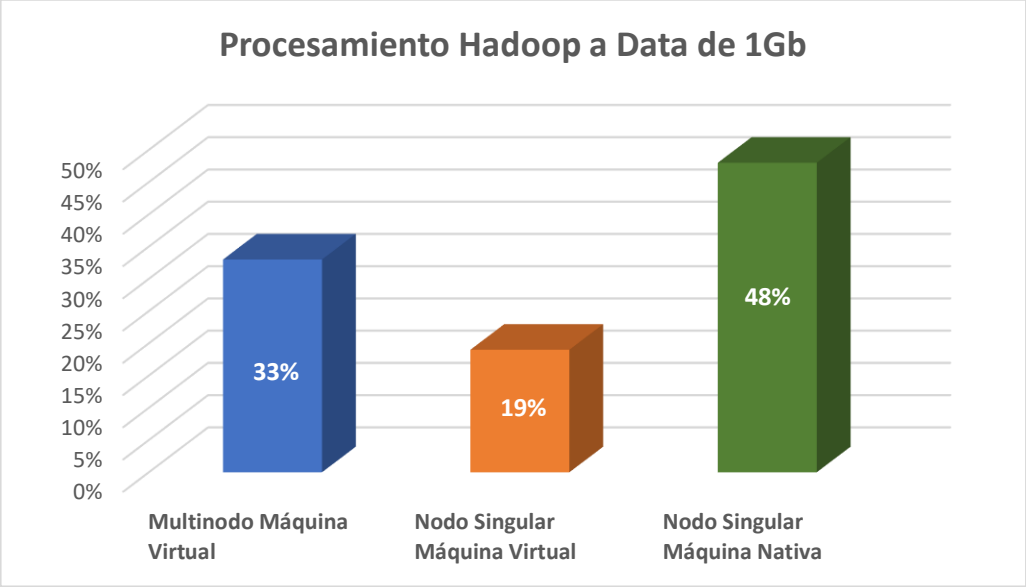


Figura 185 - Resultado Procesamiento 2Gb
Fuente: Elaboración Propia

Los resultados presentados en la figura 185, se pueden interpretar que el procesamiento de la data es mucho más veloz en una arquitectura de nodo singular en una máquina nativa, ya que el equipo tiene mayor poder de procesamiento y velocidad que una máquina virtual, pero en comparación a la prueba en equipos virtuales se demuestra que el procesamiento es mucho más veloz en una arquitectura multinodo en comparación a una de nodo singular.

La máquina nativa realiza el procesamiento de 1Gb de datos en un promedio de 31 segundos a 50 segundos, mientras que la arquitectura multinodo en máquinas virtuales lo realiza en un promedio de 50 segundos a un minuto 20 segundos y la arquitectura de nodo singular en máquina virtual lo realiza en un promedio de un minuto 20 segundos a un minuto 40 segundos.