UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN INFORMÁTICA

Análisis y uso de las herramientas de Hadoop para procesar Big Data

TRABAJO DE TITULACIÓN

AUTOR: Imba Aranda, Diego Javier

DIRECTOR: Tenesaca Luna, Gladys Alicia, MSc

CENTRO UNIVERSITARIO QUITO

2017

2017

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Master.
Gladys Alicia Tenesaca Luna.
DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Análisis y uso de las herramientas de Hadoop para procesar Big Data, realizado por Imba Aranda Diego Javier, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, Septiembre del 2017

f) . . . . . .

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, Imba Aranda Diego Javier declaro ser autor del presente trabajo de titulación: Análisis y uso de las herramientas de Hadoop para procesar Big Data, de la Titulación de Ingeniero en Informática, siendo la MSc. 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 la 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: Imba Aranda Diego Javier

Cédula: 1720752110

DEDICATORIA

El presente trabajo de titulación está dedicado a los seres más amados que tengo a mi lado. A Dios que nunca me desamparó y estuvo conmigo en cada paso para que pueda culminar esta etapa de mi vida.

A mi madre, por ser el soporte y apoyo incondicional en todas las actividades que he realizado hasta el día de hoy, ya que me ha dado un ejemplo de perseverancia, lucha y voluntad día a día con su trabajo y su esfuerzo.

Finalmente, a mi esposa por estar conmigo y motivándome constantemente con sus palabras de aliento y apoyándome noche y día para lograr las metas trazadas; y mis hijos, que son el motivo de mi vida que se convirtieron junto a mi esposa en la mayor fuente de inspiración para alcanzar logros importantes como este.

AGRADECIMIENTO

Agradezco a mi madre por haberme dado el mejor ejemplo de lucha, perseverancia y amor, acompañándome constantemente con sus consejos para convertirme en un hombre de bien.

A mi amada esposa Nelly por su amor y apoyo absoluto en toda mi etapa de formación académica, por ser el eje de mi vida que me motivó a no desmayar y alentándome hasta alcanzar mis objetivos.

A mis hijos Lorena y Joaquín por ser la base y pilar donde construyo mis sueños.

Por último mi más sincero agradecimiento y gratitud a mi tutora MSc. Gladys Alicia Tenesaca Luna por su aporte y colaboración para la realización de este trabajo de titulación.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El presente trabajo de investigación se lo realiza tomando de base Hadoop como núcleo central del procesamiento de Big Data. Dentro del ecosistema de Hadoop existen varias herramientas Open Source que facilitan la manipulación y uso de grandes volúmenes de datos, y se ha seleccionado sus 4 principales herramientas las cuales han sido agrupadas en dos casos de estudio: Flume con Hive para el caso de estudio 1, y Sqoop con Pig para el caso de estudio 2.

En ambos casos de estudio, se ha seleccionado para análisis los datos generados en la red social Twitter durante la primera y segunda vuelta electoral del 2017. Estos datos generados han sido obtenidos, almacenados, procesados y analizados para cumplir con las características que forman parte de la información que es considerada Big Data. Las herramientas seleccionadas han sido evaluadas en su arquitectura, instalación, uso y funcionalidad para diseñar un prototipo de usabilidad el cual agrupa la funcionalidad de las 4 herramientas de Hadoop; esto con el fin de facilitar su uso al usuario mediante un solo aplicativo entendible y fácil de manejar.

PALABRAS CLAVES: Big Data, Hadoop, Flume, Hive, Sqoop, Pig**,** Twitter, MySQL, Tweets, Prototipo.

ABSTRACT

The present work of investigation realizes it taking of base Hadoop as central core of Big Data's processing. Inside Hadoop's ecosystem there are several Open Source tools that facilitate the manipulation and use of big volumes of information, and there have been selected his 4 principal tools which have been grouped in two cases of study: Flume with Hive for the case of study 1, and Sqoop with Pig for the case of study 2.

In both cases, the data generated in the social network Twitter during the first and second round of elections in 2017 has been taken as the basis of analysis. This generated information has been obtained, stored, processed and analyzed to expire with the characteristics that form a part of the information that is considered to be Big Data. The selected tools have been evaluated in his architecture, installation, use and functionality to design a prototype of result which groups the functionality of 4 Hadoop's tools; this in order to facilitate its use to the user by means of a single application understandable and easy to handle.

KEYWORDS: Big Data, Hadoop, Flume, Hive, Sqoop, Pig, Twitter, MySQL, Tweets, Prototype.

INTRODUCCIÓN

Debido a la generación de grandes volúmenes de datos (Big Data) que se crean cotidianamente, se ha hecho necesario que esta información de datos resultante se le brinde un análisis, uso, valor y aporte que le genere a una persona, organización empresa, etc. El análisis de datos puede realizarse con herramientas de software de uso común como los gestores de base de datos tradicionales (Microsoft SQL Server, MySQL, etc.). Sin embargo, al tener fuentes de datos no estructurados que serán utilizados dentro de Big Data, no encajan en los almacenes de datos tradicionales.

En base a esto, una nueva clase de tecnología para el manejo de grandes volúmenes de datos ha surgido y está siendo utilizado en muchos análisis de Big Data. Esta tecnología es Apache Hadoop, el cual es un software Open Source compuesto por varias herramientas y que soporta el procesamiento de grandes volúmenes de datos (estructurados, no estructurados, semi-estructurados) a través de sistemas en un clúster. El manejo de las herramientas de Hadoop permitirá darle uso y valor a los datos obtenidos en un dominio. Es por tal motivo, que se hace necesario proporcionar información, uso y análisis de las herramientas más sobresalientes que pueden ser usadas dentro de Hadoop para el manejo de Big Data.

En los diferentes capítulos de este trabajo de investigación se ha documentado información referente a Hadoop y sus 4 principales herramientas: Flume, Hive, Sqoop y Pig. Las herramientas seleccionadas han sido evaluadas en su arquitectura, instalación, uso y funcionalidad con el fin de diseñar un prototipo de usabilidad el cual agrupa la funcionalidad de las 4 herramientas de Hadoop demostrando su uso a través de pruebas comparativas con datos obtenidos en la red social Twitter y que han sido generados con durante la primera y segunda vuelta electoral 2017.

Al finalizar se realiza un análisis e interpretación de los resultados obtenidos durante cada una de las vueltas electorales 2017, así como también un análisis comparativo general, funcional y de uso de las 4 herramientas seleccionadas de Hadoop.

De esta manera se ha llegado a cumplir con los objetivos propuestos en el presente trabajo de titulación, iniciando por las conceptualizaciones más importantes de Hadoop, hasta llegar al uso, análisis, selección y determinación de un prototipo de usabilidad en base a las 4 herramientas más sobresalientes de Hadoop usadas en las organizaciones empresariales.

CAPÍTULO 1: FORMULACIÓN DEL PROBLEMA

1.1. Antecedentes

En los últimos años debido al avance de la tecnología, el intercambio de información ha ido en un gran crecimiento el cual se los evidencia en grandes volúmenes de datos, haciéndose necesario tomar en consideración herramientas y medios de almacenamientos que permitan recolectar toda esta información para su procesamiento.

El ser humano se ha convertido en un actor dinámico que aporta en la generación de gran cantidad de información haciendo uso de dispositivos tecnológicos que han permitido la generación, transferencia, intercambio de datos. Con el pasar del tiempo, se ha hecho necesario que esta gran información de datos resultante se le asigne un nombre para su análisis, uso, valor y aporte que brinda a una persona, empresa, organización, etc.

Se denomina Big Data al análisis, captura, transformación de datos, búsqueda, intercambio, almacenamiento, transferencia, visualización y privacidad de enormes volúmenes de datos que no son tratados o presentados de manera convencional, debido a que superan los límites y capacidades de las herramientas de software que se utilizan comúnmente para la captura, gestión y procesamiento de datos. Big Data a menudo hace referencia a la utilización de análisis predictivo, búsqueda de patrones u otros métodos avanzados para extraer información o valores que un gran conjunto de datos heterogéneos.

El manejo de toda la información contenida dentro de Big Data debe realizarse a través de herramientas de software que permiten manipular los datos, para obtener resultados de análisis que aporten a las personas o a las organizaciones que ven como un valor agregado el explotar Big Data. Dentro de las herramientas de software para el procesamiento de Big Data tenemos Apache Hadoop, que cuenta con una amplia gama de gestores de procesamiento de información para la manipulación de grandes volúmenes de datos.

1.2. Justificación

En la actualidad, el análisis de Big Data ayuda a las empresas u organizaciones a tomar las mejores decisiones de negocio al permitir procesar grandes volúmenes de datos transaccionales, así como también otras fuentes de datos no transaccionales. La información obtenida proporcionará ventajas competitivas frente a organizaciones rivales beneficiando al negocio. Por ejemplo: marketing más efectivo, entender el perfil, necesidades y sentir de sus clientes respecto a los productos y/o servicios vendidos, entre otros beneficios.

El análisis de Big Data se lo realiza con herramientas de software de uso común como los gestores de base de datos tradicionales (Microsoft SQL Server, MySQL, etc.). Sin embargo, al

tener fuentes de datos no estructurados que son utilizados dentro de Big Data, no encajan en los almacenes de datos tradicionales y no son capaces de manejar la necesidad y demanda de procesamiento de grandes cantidades de datos.

Como resultado, una nueva clase de tecnología ha surgido y está siendo utilizada en muchos análisis de grandes volúmenes de datos. Esta tecnología es Apache Hadoop, el cual es un software Open Source compuesto por varias herramientas que soportan el procesamiento de grandes volúmenes de datos (estructurados, no estructurados, semi-estructurados) a través de un solo ordenador o de sistemas en un clúster.

Por lo antes mencionado, es necesario realizar la instalación y uso de las herramientas de Hadoop para proceder a realizar un análisis de todos aquellos factores que contribuyen y diferencian en el uso y procesamiento de grandes volúmenes de información.

El presente trabajo de titulación no está enfocado a trabajar para beneficio de una sola empresa u organización, ya que el análisis de las herramientas de Hadoop que son objeto de estudio, lanzarán resultados que se pueden observar por cualquier empresa que utilice o trabaje con Big Data para tomar una decisión de que herramienta se ajusta más a sus necesidades. Actualmente, en los proyectos de Big Data gobierna la tecnología Open Source, especialmente la relacionada con Apache Hadoop. Es por tal motivo, que se hace necesario proporcionar información, uso y análisis de las herramientas más sobresalientes que son usadas dentro de Hadoop para el manejo de Big Data.

1.3. Objetivos

1.3.1. Objetivo General

Analizar las 4 principales herramientas del ecosistema de Hadoop haciendo uso de Big Data.

1.3.2. Objetivos Específicos

- Investigar las conceptualizaciones importantes de Big Data.
- Establecer las 4 primeras herramientas de Hadoop que son más utilizadas en las empresas para el procesamiento de Big Data.
- Identificar beneficios, ventajas y desventajas que ofrecen las herramientas seleccionadas de Hadoop para el manejo de Big Data.
- Desarrollar un prototipo de usabilidad con las 4 herramientas seleccionadas de Big Data y que forman parte del ecosistema de Hadoop.

1.4. Alcance

El presente trabajo de titulación consiste en el uso, análisis, selección y diseño de un prototipo de usabilidad en base a las 4 herramientas más sobresalientes del ecosistema de Hadoop, las cuales permiten procesar y analizar Big Data. El prototipo de usabilidad es el resultado de la comparación y determinación de los factores que contribuyen y diferencian el uso y procesamiento de grandes volúmenes de información al hacer uso de las herramientas de Hadoop seleccionadas.

La información que nos servirá de referencia para poder probar las herramientas de Hadoop, es tomada en la primera y segunda vuelta electoral de las elecciones presidenciales 2017 en la cual se incluye información estructurada y no estructurada.

Adicional, Hadoop posee una amplia gama de herramientas Open Source para poder manipular, dividir y mapear la información, pero nos enfocaremos en las que más uso se las ha estado dando en las organizaciones empresariales que utilizan o trabajan con Big Data. Por lo cual, luego de realizar una investigación a nivel empresarial, son seleccionadas las 4 principales herramientas las cuales son parte de nuestro objeto de estudio en el presente trabajo de titulación.

CAPÍTULO 2: MARCO TEÓRICO

En este capítulo se realiza la recopilación teórica de los diferentes temas que hacen relación a las nociones más importantes sobre Big Data y a las herramientas de Hadoop, tomando en cuenta varios postulados y conceptos de diferentes autores e investigadores.

2.1. Big Data

En la actualidad, existen varios conceptos que dan una definición a lo que es Big Data, de las cuales se hacen mención y se toman en cuenta los temas más relevantes para una mejor comprensión.

2.1.1. Definición de Big Data

Uno de los términos de Big Data es presentado por (Manyika, J y otros, 2011) en donde se la define como el conjunto de datos cuyo tamaño es considerable en relación a la capacidad de captura, almacenado, gestión y análisis de las herramientas de base de datos.

Big Data según (Salvador, 2014) es la "disponibilidad de grandes cantidades de información en formatos estructurados y desestructurados en tiempo real".

Otra definición más completa es dada por la empresa consultora Gartner Inc. (2012) en donde indica que Big Data "son activos de información caracterizados por su alto volumen, velocidad y variedad, que demandan soluciones innovadoras y eficientes de procesado para la mejora del conocimiento y toma de decisiones en las organizaciones."

Es necesario mencionar los resultados de un estudio realizado por parte del Institute for Business Value y la Escuela de Negocios de Saïd en la Universidad de Oxford, en la que varios profesionales seleccionaron dos características de Big Data en base a una encuesta. Como resultado se tiene la siguiente información mostrada en la figura 1.

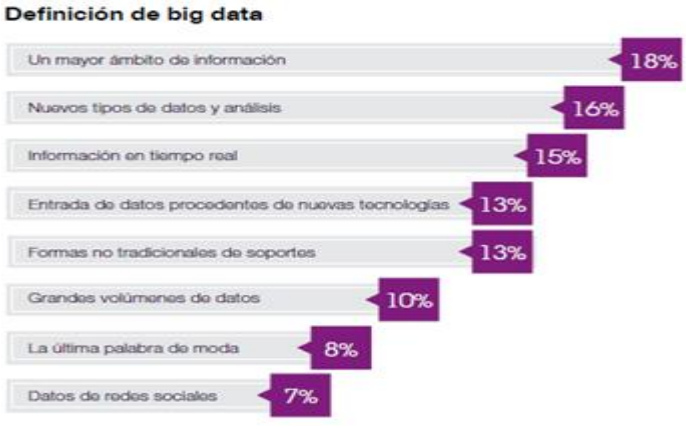


Figura 1: Definición de Big Data
Fuente: IBM Institute for Business Value

De los resultados mostrados en la figura 1, se observa que no existe una característica que domine el término de Big Data, ya que de la mayor parte de opciones elegidas por los encuestados arrojaron opiniones que relacionan a Big Data con una nueva tecnología como producto de la obtención de grandes cantidades de información, y en menor cantidad se piensa que Big Data está relacionada con los datos de las redes sociales.

Como resultado y dando una definición formal, Big Data es el análisis, procesamiento y almacenamiento de grandes volúmenes de datos que se originan de diferentes fuentes conformando un conjunto de datos estructurados y no estructurados que generan un valor para la toma de decisiones.

2.1.2. Características de Big Data

Para que los grandes volúmenes de datos sean considerados Big Data, es necesario que lleguen a cumplir una o más características que son parte de las citadas 5 Vs de Big Data:

- Volumen.
- Velocidad.
- Variedad.
- Veracidad.
- Valor.

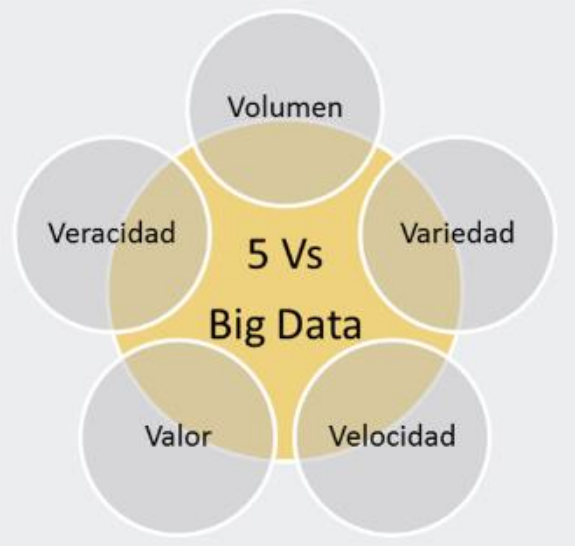


Figura 2: Las 5 Vs de Big Data
Fuente: Slocum, M. (2011).

Las 5 Vs se describen de la siguiente manera:

Volumen: según (Schroeck, M y otros, 2012) el volumen en Big Data se la denomina como: “las cantidades masivas de datos que las organizaciones intentan aprovechar para mejorar la toma de decisiones en toda la empresa”.

El volumen hace referencia a la cantidad de información que se están generando cada segundo. Esta información está en constante crecimiento debido a las diferentes fuentes en las que se generan los datos y con el pasar del tiempo los volúmenes de datos seguirán en aumento. Como ejemplo podemos mencionar la cantidad que genera continuamente las redes sociales, correos electrónicos, internet, fotos, clips de videos, sensores, etc.

Velocidad: según Laney (2001) hace referencia a que tan rápido la data es procesada. Por ejemplo, mucha de la información que se encuentra en la redes sociales se viraliza en cuestión de minutos generándose nuevos datos y moviéndose de un lado a otro. La velocidad de las transacciones de tarjetas bancarias en las que se procesa una consulta es otra manera clara de cómo la velocidad nos permite analizar de manera casi instantánea los datos sin ser necesario ubicar la información en una base de datos, proporcionándonos resultados de manera rápida para convertirla en información útil.

Variedad: en la variedad se obtienen diferentes tipos de datos que son utilizados, tales como: los datos estructurados, semi-estructurados y no estructurados.

Anteriormente, la forma de consultar la data se lo hacía a través de datos estructurados los cuales se adaptaban fácilmente a tablas o bases de datos relacionales (hojas de cálculo, bases de datos SQL, plantillas, etc.), pero en la vida real muchos de los datos no poseen una estructura definida y no se ubican en tablas o bases de datos convencionales. A estos datos se los ha considerado como datos semi-estructurados (servidores web, CDR, etc.) y datos no estructurados (correos electrónicos, fotos, audio, etc.).

Veracidad: en la veracidad se busca que los datos obtenidos en la variedad tengan un grado de confiabilidad debido a que llevan un factor de incertidumbre que son representativos en ciertos tipos de datos, tales como: factores económicos, sentimientos humanos, datos de sensores, etc.

Muchos de los datos que forman parte de Big Data, son evaluados utilizando técnicas de procesamiento para resolver problemas de datos no válidos y así tomar la información que es útil y aprovechable.

Valor: se lo define como el beneficio que brindan los datos para una empresa u organización. El valor de los datos está íntimamente ligada con la veracidad, ya que entre mayor sea la veracidad de los datos, más valor tendrá la información para el negocio. Es importante mencionar que el valor también depende del tiempo que se demoren en procesar los datos, ya que el análisis de los datos tiene una vida útil; por ejemplo: el mercado bursátil para la adquisición o venta de acciones cuyo valor depende del tiempo en que se abrió la oferta o demanda.

En resumen, cuando se mencionan las características de Big Data, todas ellas hablan de información generada en grandes volúmenes que crecen constantemente, y se debe considerar todos los diferentes tipos de datos que se encuentran disponibles para presentarse como una fuente de información aprovechable y lista para ser explotada.

La gran cantidad de datos utilizados en Big Data son el resultado de la interacción del ser humano con sistemas de información, o también por dispositivos electrónicos como GPS, detectores de movimiento, sensores, etc. Esto se lo resume en la figura 3.

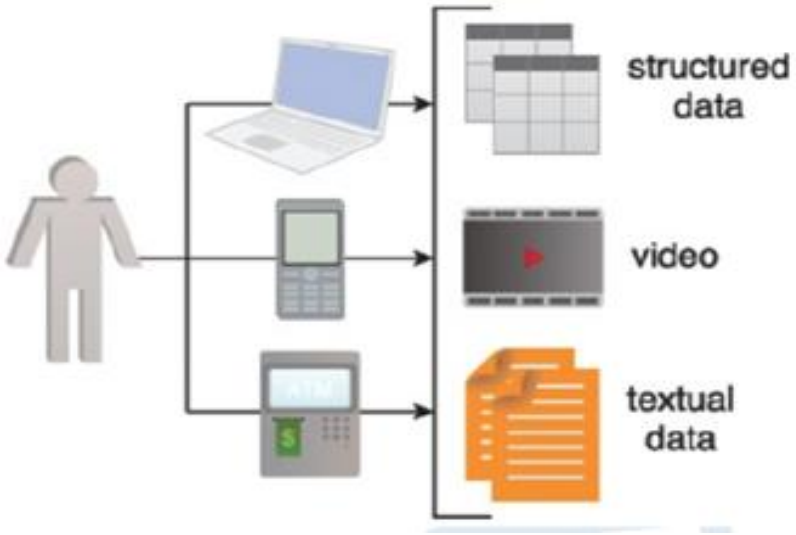


Figura 3: Tipos de datos generados por humanos o máquinas
Fuente: Universidad de Barcelona. (2015).

Para (Mayer-Schönberger & Cukier, 2013) “los datos son el nuevo petróleo”, de esta manera se resalta el potencial que tiene Big Data para la innovación tecnológica, creación de fuentes de empleo y la generación de rentabilidad para las empresas convirtiéndose en un activo principal de los modelos de negocio.

2.1.3. Ventajas de Big Data

Muchas de las ventajas que tiene Big Data están dirigidas a varias empresas que enfocan sus casos de negocio para obtener un valor añadido a la toma de decisiones necesarias en un momento clave.

Según (Schroeck, M y otros, 2012) una de las principales ventajas que aporta Big Data a una empresa es:

- Decisiones más rápidas que permiten una captura y análisis de datos en tiempo más real para respaldar la toma de decisiones en el "punto de impacto". Por ejemplo cuando un cliente está navegando por un sitio web, o la conversación telefónica que puede tener con un representante del servicio al cliente.

Es necesario mencionar otras ventajas que aporta Big Data (Mayer-Schönberger & Cukier, 2013) en las que se resalta de mejor manera las siguientes ventajas empresariales de Big Data:

- Implementación de mejoras tecnológicas que posibilitan la adquisición de datos y que permiten descubrir las necesidades y puntos de mejora en la compañía.
- El análisis de los datos mejora sustancialmente la toma de decisiones dentro de una compañía reduciendo al mínimo los riesgos. Así, algunas organizaciones ya están optimizando sus decisiones mediante el análisis de datos de clientes, empleados, o incluso sensores incorporados en los productos.
- Big Data facilita que las compañías evalúen sus productos. Mediante el análisis de datos, obtienen información muy valiosa que les permite crear nuevos productos o rediseñar los ya existentes.
- Segmentación de los clientes para personalizar acciones. De esta forma, las empresas orientan sus servicios y satisfacen las necesidades de sus consumidores de forma específica.
- Mejora la accesibilidad y la fluidez de la información dentro de la propia empresa, creando una dinámica de trabajo más rápida y eficaz.

De forma general, son varias las ventajas que posee Big Data y tienen un denominador común el cual es el procesamiento de grandes cantidades de información de forma rápida, generando un valor agregado a la organización para que tenga una ventaja competitiva y mejore su gestión empresarial.

2.1.4. Desventajas de Big Data

Las desventajas de Big Data están más relacionadas al desconocimiento de la misma y se describen a continuación:

- La principal desventaja es el costo del software y hardware que se debe usar para implementar una solución de Big Data.
- Falta de profesionales que tengan conocimiento y se encuentren capacitados para el manejo de Big Data.
- El tener un gran volumen de información no implica necesariamente que se tenga que hacer uso de un proyecto o solución de Big Data. Es preciso entender y analizar las necesidades del cliente.
- La calidad de los datos es uno de los grandes obstáculos en Big Data, ya que se tiene una gran cantidad de información pero no se la sabe entender y clasificar. No todos los datos son información y es necesario conocer si los datos a analizar tienen valor.
- Dependiendo del país, existe un marco regulatorio de protección de datos personales, el cual limita realizar el procesamiento de datos masivos en la que se incluya información sensible de las personas.

Se han mencionado algunas desventajas de Big Data, pero es necesario que una empresa analice, reflexione y considere si es necesaria la utilización de una solución de Big Data tomando en cuenta los costos de implementación, los recursos necesarios y si la información es útil para la organización. Luego de revisar los pros y los contras, y si la organización opta por un proyecto de Big Data, las ventajas serán mucho mayores que las desventajas.

2.1.5. Metodología para extraer o procesar Big Data

La forma de obtener y procesar Big Data se lo hace mediante una serie de pasos en las que se debe cumplir con actividades y tareas relacionadas a la adquisición, procesamiento, análisis y reutilización de datos,

Según la empresa capacitadora Handytec (2016) el procesamiento de Big Data se divide en las siguientes 9 etapas mencionadas en la figura 4:

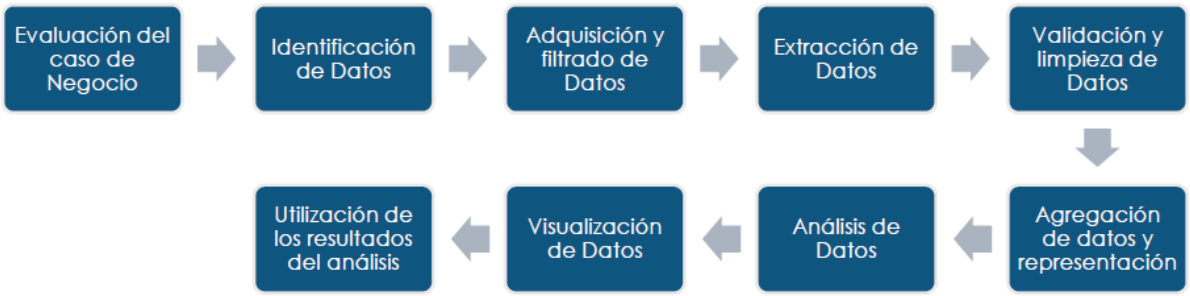


Figura 4: Las 9 etapas para el procesamiento de Big Data
Fuente: Handytec. (2016).

Las 9 etapas para la extracción y procesamiento de Big Data se describen a continuación:

Evaluación del caso de negocio: La evaluación del caso de negocio permite a los tomadores de decisiones conocer cuáles son los recursos de la empresa que deberán ser utilizados y los desafíos de negocio que tendrá el análisis de datos. Hay que tomar en cuenta el presupuesto para la adquisición de herramientas de software y hardware, así también el equipo de trabajo de las personas que trabajarán con la información de Big Data.

Luego de la evaluación, se generan los requerimientos que se plasmarán en el caso de negocio para conocer si realmente se están abordando problemas de Big Data. Es necesario recordar que el problema de negocio debe tener una o más características de Big Data, tales como: volumen, velocidad, variedad, veracidad, valor.

Identificación de los datos: en esta etapa se identifican las fuentes de donde se obtiene el conjunto de datos para el realizar el caso de negocio.

Entre mayor sea las fuentes de datos se aumenta la probabilidad de encontrar patrones ocultos y correlaciones que beneficien al caso de negocio propuesto. Dependiendo del alcance del proyecto, el conjunto de datos requeridos son de fuentes internas y/o externas a la empresa.

Los datos internos son aquellos que se obtienen de fuentes internas, tales como mercado de datos (CRM, ERP, BPM, etc.) y sistemas operativos (Excel, etc.), que generalmente se recopilan y comparan con una especificación de datos predefinida.

Figura 5: Fuentes de datos internos
Fuente: Handytec. (2016).

Al contrario, los datos externos se los obtiene de posibles proveedores de datos de terceros. Estos datos se encuentran en blogs o sitios web (redes sociales, datos georeferenciados, etc.) y son accesibles mediante la cosecha de datos utilizando herramientas automatizadas.

Figura 6: Fuentes de datos externos
Fuente: Handytec. (2016).

Adquisición y filtrado de datos: en esta etapa los datos son recolectados de las fuentes de datos internos y externos mencionados en la etapa anterior. Los datos obtenidos deben ser filtrados de manera automática para eliminar los datos corruptos o que no aportan ningún valor para los objetivos del análisis.

Dependiendo de la fuente de datos, la información recolectada viene como un conjunto de archivos o necesitan de la integración con una API (por ejemplo Twitter). En muchos casos los datos obtenidos de las fuentes externas son datos no estructurados y en el proceso de filtrado algunos de los datos irrelevantes deben ser descartados.

Extracción de los datos: en esta etapa se busca extraer los datos incompatibles para transformarlos en un formato legible para realizar el análisis de datos en Big Data.

No todos los datos requieren de extracción y transformación. Como ejemplo se tiene los documentos escaneados; la extracción del texto de estos documentos se dan mediante la lectura directa del documento en su formato nativo.

Validación y limpieza de datos: en esta etapa se busca establecer reglas de validación que muchas veces son complejas, así como también la eliminación de datos no válidos conocidos.

La validación de datos es utilizada para explorar conjuntos de datos interconectados con el fin de suplir la falta de datos válidos. Como ejemplo se tiene la figura 7, en donde se lo explica de la siguiente manera:

- El primer valor en el conjunto de datos B se valida con su valor correspondiente en el conjunto de datos A.
- El segundo valor en conjunto de datos B no se valida con su valor correspondiente en el conjunto de datos A.
- Si se pierde un valor, este se inserta a partir del conjunto de datos A.

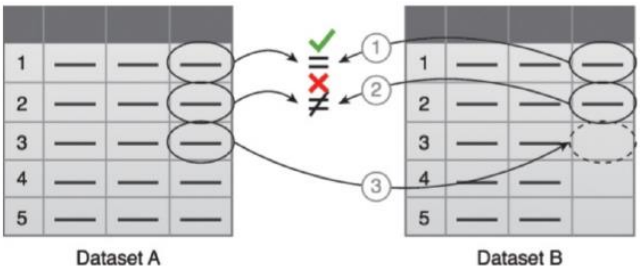


Figura 7: Validación de datos en Dataset A y B
Fuente: Handytec. (2016).

Si lo que se requiere es realizar un análisis en tiempo real, es necesario contar con un sistema complejo que trabaje en memoria para validar y limpiar los datos a medida que van llegando desde su fuente, caso contrario se lo realiza través de una operación ETL fuera de línea.

Agregación de datos y representación: es esta etapa se busca la integración de múltiples conjuntos de datos mediante un método de conciliación para llegar a tener una versión unificada. Su fin es la estandarización en una estructura de datos haciendo uso de una amplia gama de técnicas y proyectos de análisis. En la figura 8 se observa cómo se obtiene una estructura de datos estandarizada combinando el conjunto de datos A y B.

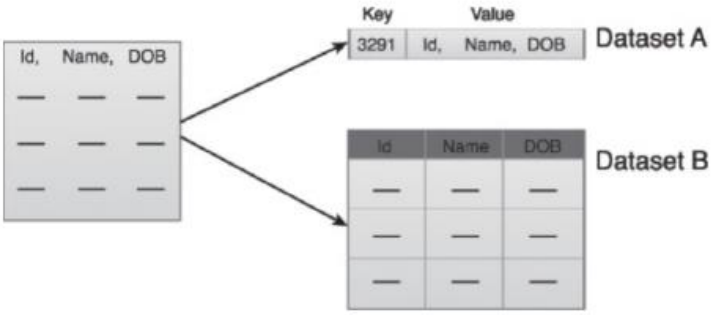


Figura 8: Estructura de datos estandarizada combinando Dataset A y B
Fuente: Handytec. (2016).

Análisis de datos: la etapa de análisis de datos realiza la tarea de análisis real de todos los datos obtenidos en la etapa anterior.

El análisis de los datos son clasificados en: análisis de confirmación y análisis exploratorio. El primero se lo realiza bajo un enfoque deductivo, el cual investiga la causa del fenómeno que se supone de antemano. Se parte de una hipótesis y luego del análisis se prueba o refuta la hipótesis para dar respuestas definitivas. En cambio, el análisis exploratorio se lo hace bajo un enfoque inductivo que se encuentra ligado estrechamente con la minería de datos. No se tiene hipótesis o suposiciones, ya que los datos son explorados a través del análisis para desarrollar una comprensión de la causa del fenómeno.

Visualización de datos: en esta etapa se busca utilizar técnicas de visualización de datos y herramientas que permitan informar gráficamente los resultados del análisis realizado para que sean interpretados de manera clara a los usuarios del negocio. Otro aspecto a tener en cuenta en esta etapa, es la inclusión de métodos simples de desagregación para que los usuarios tengan conocimiento de cómo se agregaron o se generaron los resultados presentados.

Los resultados que se presentan a los usuarios son una respuesta visual que permite el descubrimiento de soluciones a las que los usuarios no han formulado preguntas y que influyen en la interpretación de los resultados. Por lo tanto, es necesario utilizar técnicas de visualización de datos que sean las más adecuadas para no perder el dominio del contexto del negocio.

Utilización de los datos de análisis: los resultados de los análisis de datos producen nuevos “modelos” que generan conocimientos y entendimientos sobre la naturaleza de los patrones y las relaciones que existen dentro de los datos analizados. Estos modelos son abstraídos como

una ecuación matemática o un conjunto de reglas, las cuales aportan a la mejora lógica de un negocio o sirven como base para un nuevo sistema de software de la empresa.

Como ejemplo de utilización de los datos de análisis tenemos:

- Entrada para Sistemas Empresariales, donde los resultados son utilizados para la mejora y optimización de los sistemas empresariales tales como: mejorar la lógica de programación en los sistemas existentes de una empresa o también sirven de base para la consecución de nuevos sistemas.
- Optimización de Procesos de Negocio: las anomalías detectadas, patrones identificados, y correlación de los datos, son utilizados para refinar o mejorar los procesos de un negocio tales como: mejora en la producción de un producto, conocer las preferencias de un cliente, mejorar la cadena de suministro mediante la utilización de transporte, etc.

2.1.6. Datificación de Big Data

La datificación según (Mayer-Schönberger & Cukier, 2013) es la recopilación de la información de lo que existe bajo el sol y que se mide con la infraestructura tecnológica en la cultura de los datos.

En Big Data se tiene una gran cantidad de datos que se recopila para que sean procesados como un todo, y que no se analizan haciendo usos de métodos convencionales de procesamiento de datos. Es por tanto, que los datos siempre se los ha tenido de diferentes formas, pero ahora estos datos crecen constantemente y de manera muy rápida dando origen a la datificación.

Con la datificación (Mayer-Schönberger & Cukier, 2013) se busca transformar un fenómeno en un formato que se cuantifique para ser tabulado y analizado. La datificación es un resultado de la digitalización, ya que las fuentes de información son masivas y al digitarlas se hace uso de los ordenadores para que mediante algoritmos, se haga el análisis que permita entender el comportamiento humano, las tendencias y su forma de pensar en algún instante del tiempo mediante el análisis cuantitativo de textos.

Es necesario mencionar (Mayer-Schönberger & Cukier, 2013) que “una vez que se ha datificado el mundo, los usos potenciales de la información no tiene más límite que el ingenio personal”. Entre los usos de la información que se dan tenemos los comportamientos de las personas, por ejemplo: anteriormente el enviar una carta a una persona no tenía una relevancia en la

generación de datos, pero al cambiarla a un correo electrónico se ha generado un registro. Con el pasar del tiempo se va generando más registros que digitalizan aspectos de nuestra vida y crecen exponencialmente. Como resultado, podemos saber a qué destinatario se ha enviado mayor cantidad de correos electrónicos, cuantos correos se han enviado, etc.

En contexto, para (Agudelo, 2015) las fuentes de información para la datificación de Big Data son:

- **Web y Redes Sociales**, la información se la obtiene a través de los diferentes contenidos webs (fotos, foros, blogs, etc.) y las redes sociales tales como: Twitter, Facebook, LinkedIn, Flickr, Instragram, etc.
- **Biométrico**, basándose en la seguridad, seguimiento y control de las personas se obtiene información biométrica. Por ejemplo: huellas digitales, reconocimiento facial, etc.
- **Generado por Humanos**, es información que genera la interacción humana. Por ejemplo: documentos, presentaciones, correos electrónicos, notas de voz, SMS, etc.
- **Transacción de Big Data**, en la que la generación de Big Data se da por la masificación de las fuentes de datos a través de diferentes medios. Por ejemplo: registros de servicios públicos, registros médicos, reportes de llamadas telefónicas.
- **Máquina a Máquina**, esta información se la obtiene de la interacción de los diferentes dispositivos electrónicos. Ejemplo: imágenes satelitales, datos científicos (datos sísmicos, atmosféricos, etc.), GPS, entre otros.

De forma general, existen muchas fuentes de generación de datos para que exista Big Data; pero la datificación crea nuevas brechas y vías de análisis. Es importante conocer el ámbito de los datos que se posee, ya que no se debe datificar sin previamente realizar un análisis de lo que se quiere llegar a alcanzar, esto con el fin de brindar resultados de datos que lleguen a generar un valor a la organización

2.1.7. Consideraciones a tomar en cuenta para el tamaño de un Data

El concepto de Big Data se refiere a toda la información o datos que no es analizada o cuantificada utilizando métodos o procesos tradicionales. Mucha de esta información supera el procesamiento de un software de habitual, el cual no soporta el manejo y gestión de grandes volúmenes de datos. En una publicación realizada por la Universidad de Barcelona (2015), Big Data "no se no se refiere a alguna cantidad en específico, ya que es usualmente utilizado cuando se habla en términos de peta bytes y exabytes de datos".

Para poder tener una idea del tamaño de la información digital en términos de bytes (unidad básica de la información), se construye la escala de medida digital de bytes:

- Kylobyte (KB) = 103 = 1,000 bytes.
- Megabyte (MB) = 106 = 1,000,000 bytes.
- Gigabyte (GB) = 109 = 1,000,000,000 bytes.
- Terabyte (TB) = 1012 = 1,000,000,000,000 bytes.
- Peta byte (PB) = 1015 = 1,000,000,000,000,000 bytes. **(a partir de este tamaño se considera Big Data)**
- Exabyte (EB) = 1018 = 1,000,000,000,000,000,000 bytes.
- Zettabyte (ZB) = 1021 bytes.
- Yottabyte (YB) = 1024 bytes.
- Quintillón (QB)= 1030 bytes.

2.2. Hadoop

Hadoop según (Schneider, 2014) es un sistema Open Source que se ha convertido en una herramienta revolucionaria para el procesamiento de Big Data, ya que facilita el almacenamiento, procesamiento y análisis de grandes cantidades de datos utilizando nodos o servidores.

2.2.1. Descripción de Hadoop

Apache Hadoop en su sitio oficial (The Apache Software Foundation, 2017) es descrita como una solución de software libre que procesa grandes volúmenes de información de hasta valores de exabytes, en el cual los datos se encuentran distribuidos en diferentes nodos (servidores) que usan modelos de programa simples.

Hadoop según (Schneider, 2014) “aprovecha la potencia del procesamiento paralelo masivo como una ventaja de Big Data, por lo general usando varios servidores básicos”.

Otra definición de Hadoop brindada por (Pérez & Asturiano, 2015) “es un sistema distribuido Open Source que pertenece a Apache Foundation diseñado enteramente en Java para almacenar y procesar grandes volúmenes de información,... posee dos componentes: HDFS y MapReduce además de varios “frameworks” y “apps” que giran alrededor de ellos para complementarlo y reforzarlo.”

Es necesario mencionar que Hadoop no es programa o aplicación que se descarga y se instala directamente bajo un ordenador o un servidor. Hadoop se conforma de un ecosistema de proyectos que se distribuyen bajo Apache Software Foundation; este conjunto de proyectos aporta una serie de funcionalidades adicionales que parten de HDFS y MapReduce. El tema de ecosistema de Hadoop se lo describirá en la sección 2.2.3.

2.2.2. Características de Hadoop

Las principales características de Hadoop son:

- Escalabilidad (White, 2015): Hadoop es linealmente escalable, su arquitectura distribuida permite funcionar en varios nodos (servidores) formando un clúster. Su funcionamiento se basa en un nodo máster y el resto de nodos en esclavos. Hadoop tiene la flexibilidad de poder agregar nodos al clúster de manera fácil y sencilla, de manera que se vuelve escalable a cualquier cambio en la variación de los datos que se vaya a procesar.
- Alta disponibilidad (White, 2015): los ficheros de Hadoop, mediante una variable de configuración, se replican varias veces con el fin de brindar un sistema de fiable.
- Tolerancia a fallos (White, 2015): Hadoop realiza una copia automática de los datos en los nodos; para que cualquier falla o caída de uno o varios nodos no afecten al funcionamiento del sistema.
- El licenciamiento de Hadoop es Open Source, pero dependiendo de las funcionalidades y recursos que quiera optar el negocio o la empresa, se hace uso de las distribuciones del ecosistema de Hadoop que han sido adaptadas por diferentes fabricantes.
- Una de las principal características de Hadoop es el procesamiento y gestión de altos volúmenes de datos, estructurados, semi-estructurados y especialmente no estructurados que se encuentran basados en archivos.
- Hadoop no funciona sin MapReduce.
- El procesamiento, gestión y almacenamiento de datos en Hadoop son fáciles de depositar en archivos y copiarlos en el HDFS.
- Los datos en Hadoop se encuentran de forma distribuida, al realizar una búsqueda de información entre los nodos se la realiza de forma rápida, ya que se accede de forma paralela a la información.

2.2.3. Ecosistema y herramientas de Hadoop

El ecosistema de Hadoop está compuesto por varios proyectos o iniciativas Open Source que modifican o complementan el núcleo (MapReduce y HDFS) de Hadoop. Este ecosistema de Hadoop se lo muestra en la figura 9.

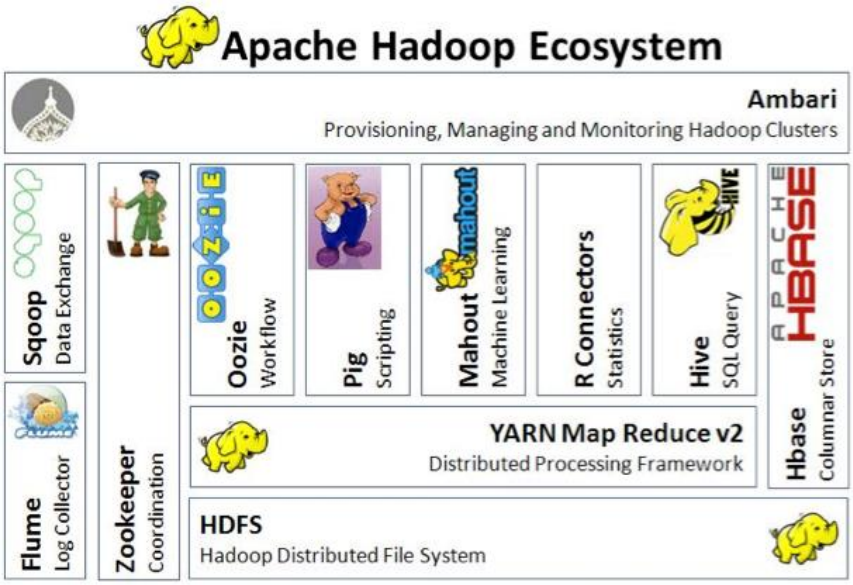


Figura 9: Ecosistema de Hadoop
Fuente: Agudelo C. (2015).

Como se observa en la figura anterior, Hadoop se ha convertido en la parte fundamental de muchos proyectos y sistemas en las organizaciones. Constantemente se generan y liberan nuevos proyectos o herramientas de tecnología que aportan al funcionamiento de Hadoop.

Según (Hurwitz & otros, 2013) estas son las principales herramientas que conforman el ecosistema de Hadoop:

- HDFS (Hadoop Distributed File System).
- Ambari.
- HBase.
- Hive.
- Sqoop.
- Pig.
- ZooKeeper.
- Mahout.
- Lucene/Solr.

- Avro.
- Oozie.
- Flume.

De forma general y resumida (Hurwitz & otros, 2013) son descritas las herramientas a continuación:

HDFS (Hadoop Distributed File System): también es conocido como Sistema de Archivos Distribuido Hadoop, el cual está diseñado para almacenar y dividir grandes muestras de datos entre varios nodos que se recuperan en el caso que se presente la falla de alguno de los nodos. El sistema de archivos soporta la tolerancia a fallos bajo un alto rendimiento.

Ambari: posee una interfaz web gráfica para el usuario con el objetivo de permitir una mejor gestión de Hadoop, facilita el uso para el suministro, supervisión y administración de los clústeres de Hadoop. Ambari está diseñado con una arquitectura de tipo servidor – agente. El servidor negocia con los agentes para llevar a cabo tareas como la instalación de nuevos servicios y la gestión de agrupación.

HBase: se caracteriza por ser una de las bases de datos de Hadoop. HBase permite el almacenamiento y búsqueda de grandes volúmenes de datos para que el MapReduce de Hadoop se ejecute en tiempo real. No sigue un esquema relacional, por lo cual no utiliza un lenguaje estructurado como el de SQL.

Hive: se encarga de regularizar el proceso de extracción de los bits de todos los archivos que se generan en HBase. Hive presenta métodos de consulta de los datos usando un lenguaje similar al SQL, llamado HiveQL.

Adicional, Hive hace uso del MapReduce de Hadoop cuando el rendimiento no es el correcto. Tiene interfaces JDBC/ODBC, por lo que empieza a funcionar su integración con herramientas de Inteligencia de Negocios.

Sqoop: es una herramienta basada en líneas de comandos, la misma que es diseñada para transferir grandes volúmenes de datos entre Hadoop y sistemas de almacenamiento (MySQL u Oracle) con datos estructurados, traduciendo las tablas en una combinación configurable para HDFS, HBase o Hive.

Pig: facilita a los usuarios de Hadoop el poder realizar el análisis de datos centrándose en un lenguaje procedural de alto nivel y enfocándose en menor manera en la creación de programas

MapReduce. Pig está orientado para trabajar con cualquier tipo de dato y se compone de un lenguaje propio llamado Pig Latín y un entorno propio de ejecución.

ZooKeeper: es un servicio centralizado que se encarga de mantener la información de prestación de servicios que son utilizados en las aplicaciones distribuidas, manteniendo requerimientos comunes que se necesiten en grandes entornos de clústeres. Algunos ejemplos de estos objetos son: información de la configuración, jerarquía de nombres, entre otros.

Mahout: es un proyecto de Hadoop que se fundamenta en la creación de un aprendizaje automático y minería de datos usando Hadoop. En otras palabras, Mahout ayuda a descubrir patrones en grandes volúmenes de datos. Posee algoritmos propios de recomendación, clustering y clasificación.

Lucene/Solr: es una librería escrita en Java que se utiliza para la gestión de textos distribuidos. Lucene permite indexar cualquier texto, que posteriormente son encontrados utilizando cualquier criterio de búsqueda. Aunque Lucene sólo funciona en texto plano, hay plugins que permite la indexación y búsqueda de contenido en documentos Word, Pdf, XML o páginas HTML.

Avro: es un sistema de serialización de datos mediante texto plano, JSON, XML o formato binario. La gran cantidad de datos que posee Hadoop son serializados para que sean procesados y almacenados de manera que el rendimiento de tiempo sea efectivo. Avro es optimizado para minimizar el espacio de disco necesario para los datos, incluso los datos son leídos fácilmente desde diferentes lenguajes de programación.

Oozie: permite la gestión de un flujo de trabajo de forma secuencial para que forme una unidad lógica de trabajo. El flujo de trabajo de Oozie está especificado como un gráfico dirigido cíclico en la que se indica la serie de secuencia a ejecutarse. Ozzie es de ayuda para otros proyectos de Hadoop tales como: MapReduce, Pig, Hive y Sqoop.

Flume: es un proyecto de Hadoop que se encarga de la recolección de grandes volúmenes de datos de forma eficaz y rápida desde diferentes servidores, y son almacenados en un solo repositorio central de datos. De esta manera se facilita la recolección de datos los cuales quedarán listos para su análisis.

Los proyectos que forman parte del ecosistema de Hadoop, permiten combinarse entre sí y crear asociaciones debido a sus diferentes funcionalidades.

2.2.4. Arquitectura de Hadoop

La arquitectura de Hadoop según (White, 2015) se basa en tres componentes fundamentales y básicos mostrados en la figura 10:

- **HDFS:** HDFS (Hadoop Distributed File System) es un sistema de ficheros distribuido el cual trabaja con grandes flujos de datos a través de la lectura y escritura de ficheros. La disponibilidad y escalabilidad son beneficios adicionales que posee HDFS, debido a la replicación de los datos y tolerancia a fallos.
- **Hadoop MapReduce:** MapReduce es el motor de Hadoop, el cual es creado para el proceso distribuido de los datos. MapReduce paraleliza el trabajo, separando la complejidad que existe en los sistemas distribuidos.
- **Hadoop Commons:** es un conjunto de utilidades sobre las cuales se integran a Hadoop como subproyectos. Este conjunto de proyectos han sido mencionados en el punto 2.2.3 del presente trabajo.
- **Otras partes a mencionar:**
    - DataNodes.
    - SecondaryNameNode.
    - Balanceador.
    - JobTracker y Tasktraker.

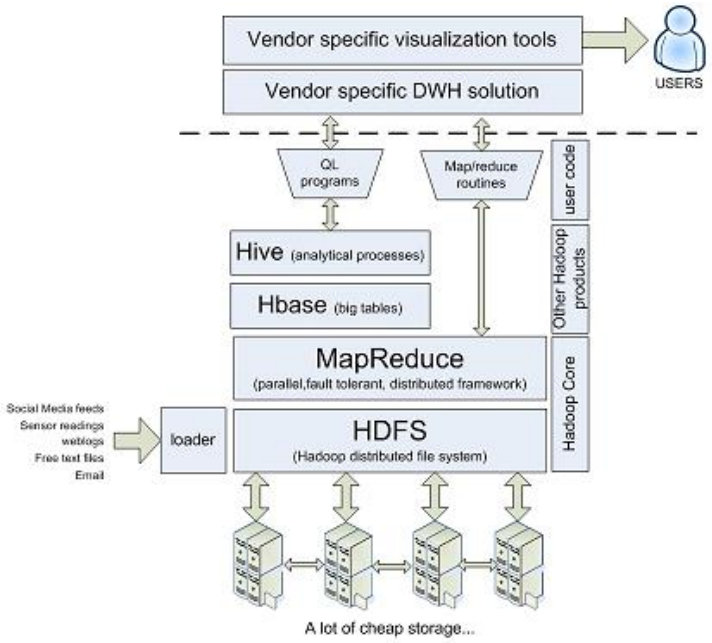


Figura 10: Arquitectura de Hadoop
Fuente: White T. (2015).

2.2.5. Funcionamiento de Hadoop

Hadoop se va a dividir en dos partes para un mejor entendimiento: en MapReduce y HDFS.

MapReduce es el alma de Hadoop, se basa en un modelo de sistema de programación para el procesamiento distribuido de grandes volúmenes de datos. El funcionamiento se basa en dos fases: fragmentar los datos y reducirlos para dar solución al problema. Un ejemplo es una función Y que se va a aplicar sobre un conjunto de datos A, A se fragmenta en pequeños pedazos y se aplica la función Y (esta es la fase Map) y el resultado de todos los "Maps" es agrupado por Reduce convirtiéndolo en un resultado único (esta es la fase Reduce). El trabajo del MapReduce es en paralelo, por tal motivo los datos son divididos en los *n* nodos con *m* tareas (a esto se llama "tasktracker"), y se da preferencia a los datos de los nodos en el que se está procesando el dato ("data locality").

HDFS se convierte en el cuerpo de Hadoop, en el sistema de archivos distribuido de Hadoop (HDFS) los datos son replicados y divididos en los diferentes clústeres de discos los mismos que tienen los nodos para que entre a funcionar MapReduce. HDFS facilita a MapReduce el procesamiento de archivos de hasta 10 Terabytes de tamaño y accede a grandes flujos de datos a través de la lectura haciendo uso del *Streaming.*

2.2.6. Selección de las 4 principales herramientas de Hadoop utilizadas por las organizaciones empresariales para el procesamiento de Big Data

Dentro del ecosistema de Hadoop, existen varias herramientas que se ha convertido en la parte fundamental de muchos proyectos y sistemas en las organizaciones empresariales. A continuación se menciona a las empresas que han optado por Hadoop como un eje central para el procesamiento de grandes volúmenes de datos:

- **Facebook:** basándonos en una publicación realizada por un ingeniero que labora en Facebook (Sen, 2008, citado en White, 2015, p.567), un buen ejemplo de manejo de grandes volúmenes de datos es Facebook. La información que se genera constantemente a través de sus varios servicios (web, publicidad, mensajería, videollamada) implica una compleja administración de millones de datos, pero el desafío principal está en la disponibilidad instantánea de la información para los usuarios utilizando un desarrollo basado en Apache Hadoop. Un ejemplo de esto es Facebook Messages, la cual es la primera aplicación orientada al usuario construida sobre la plataforma Apache Hadoop. Apache HBase es una capa de base de datos construida sobre Hadoop diseñada para soportar billones de mensajes por día. Adicional, en

Facebook se utiliza Apache Hadoop en tres tipos de sistemas: como un almacén para análisis web mediante Apache Pig, como almacenamiento para una base de datos distribuida con Apache Hive y para copias de seguridad de base de datos MySQL utilizando Apache Sqoop.

- **NASA:** en la revista de la NASA (Administración Nacional de la Aeronáutica y del Espacio) se menciona que utilizan Hadoop (Schnase & otros, 2012) recibiendo varios petabytes de información diariamente, los mismos que son enviados desde todos los satélites y misiones que tienen en el espacio. Para el manejo de estos volúmenes de datos, la NASA hace uso de varios métodos para el filtrado de la información válida, y que posteriormente es procesada y analizada. Por ejemplo, todo lo que proviene de los Sistemas de Observación de la Tierra son procesados, archivados y repartidos por el Centro Activo de Archivos Distribuidos. Para el procesamiento de la información la NASA hace uso del software libre por su bajo costo. Este software es Apache Hadoop e implementa una herramienta que se llama Apache TIKA que sirve para extraer metadatos y texto estructurado de los documentos. Apache TIKA es un proyecto desarrollado por la comunidad Apache el cual combina las funcionalidades de Apache Hive, Apache Pig y HBase.
- **Cloudera Inc.:** otro ejemplo de uso de Hadoop por una organización se encuentra en un artículo publicado en la Universidad de Mondragón (2014), en la que menciona a Cloudera como una compañía que proporciona soporte, servicios y software basado en Apache Hadoop. Posee una distribución que utiliza muchos de los aspectos y funcionalidades del ecosistema de Hadoop, pero también presenta muchas mejoras. Cloudera ha implementado una serie de características en su producto, desde una herramienta de gestión y monitoreo llamada Cloudera Manager, hasta un motor SQL para ejecutar datos relacionales sobre Hadoop llamado Impala. Cuando los clientes de Cloudera necesitan algo que no posee Hadoop, lo construyen; de esta manera se busca innovar rápidamente para satisfacer las demandas de los clientes y sobresalir con una solución que les diferencia a las de otros proveedores. Algunos de los clientes de Cloudera tienen más de 1 petabyte de información bajo gestión a través de varios nodos. Entre las funcionalidades que tomó Cloudera de las herramientas de Hadoop se tiene: HBase, propiedades Flume, comandos Hive, scripts Pig, flujos Oozie, trabajos Spark y trabajos Sqoop.
- **Amazon Web Services (AWS):** en una publicación realizada por Cloudera, Inc (2017) menciona a Amazon Web Services como una distribución de servicios de computación

en la nube (servicios web) que en conjunto forman una plataforma de nube pública alojada de Hadoop, y que son ofrecidas a través de Internet por Amazon.com. Amazon Elastic Map Reducer (EMR) es el producto de Hadoop que la compañía utiliza en AWS para ofrecer servicios de gestión de Big Data. Amazon EMR provee de un servicio administrado y manejable que hace que sea rápido, fácil y rentable el ejecutar Apache Hadoop y Spark para procesar grandes cantidades de datos. Adicional, soporta potentes y eficientes herramientas de Hadoop tales: como Flume, Sqoop, Hive, Pig y más.

Para resumir la utilización de las herramientas de Hadoop por las empresas antes mencionadas se ha creado la tabla 1:

Tabla 1: Herramientas Hadoop seleccionadas

Empresa	Servicios	Sistemas	Herramientas Hadoop Usadas
Facebook	- Web. - Publicidad. - Mensajería. - Video llamada.	- Facebook Messages. - Almacén de datos para análisis web. - Almacenamiento de una base de datos distribuida. - Copias de seguridad de base de datos.	- Hbase. - Pig. - Hive. - Sqoop.
NASA	- Observación de misiones y satélites en el espacio.	- Centro Activo de Archivos Distribuidos. - Apache TIKA para extraer metadatos y texto estructurado de documentos.	- Hive. - Pig - HBase.
Cloudera Inc.	- Soporte. - Servicios. - Software basado en Apache Hadoop.	- Cloudera Manager. - Impala.	- Hbase. - Flume. - Hive. - Pig. - Oozie. - Spark. - Sqoop.
Amazon Web Services (AWS):	- Servicios de computación en la nube de Amazon. - Servicios de gestión de Big Data en la nube.	- Amazon Elastic Map Reducer (EMR).	- Flume. - Sqoop. - Hive. - Pig.

Fuente: Elaboración propia.

Al revisar las herramientas de Hadoop que han sido usadas en algunas compañías, se ha determinado que las principales herramientas utilizadas en proyectos y servicios son: **FLUME, SQOOP, PIG y HIVE.**

CAPÍTULO 3: USO DE LAS 4 HERRAMIENTAS SELECCIONADAS DE HADOOP

3.1. Casos de estudio combinando herramientas de Hadoop

Las 4 herramientas de Hadoop seleccionadas son agrupadas de acuerdo a su funcionalidad para realizar los pasos de recolección, almacenamiento y tratamiento de datos. Por lo cual, es necesario hacer uso de 2 casos de estudio en los que se utiliza las herramientas de Hadoop las cuales son evaluadas mencionando su arquitectura, instalación (ANEXOS), uso y funcionalidad con el fin de diseñar un prototipo de usabilidad que es desarrollado en el apartado 4.4 del presente trabajo.

Para los 2 casos de estudio se utiliza el sistema operativo Linux Ubuntu versión 14.04 LTS. Este sistema operativo está instalado bajo una máquina virtual (Oracle VM VirtualBox) y posee las siguientes características:

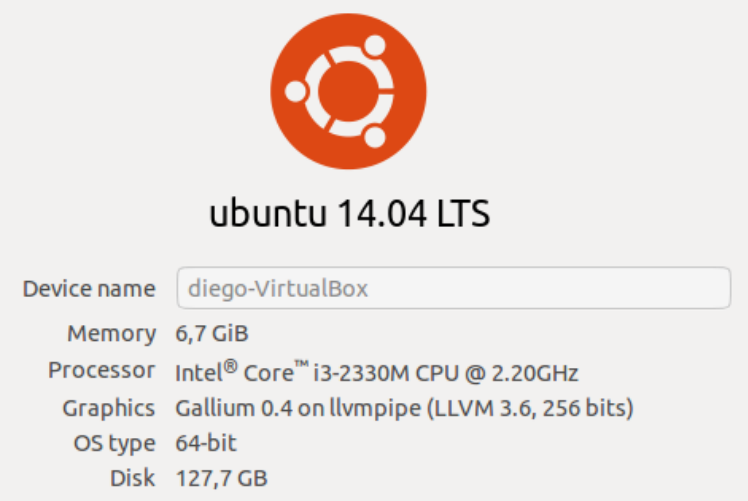


Figura 11: Configuración de Ubuntu en Máquina Virtual
Fuente: Elaboración propia.

3.1.1. Caso de estudio 1

El primer caso de estudio se centra en la descarga de datos relacionados a la primera vuelta de las elecciones presidenciales del 2017 de la red social Twitter, ya que es la red social que proporciona más facilidades con menos restricciones al momento de obtener información. La búsqueda de temas por palabras clave ayuda mucho al momento de establecer patrones de información y sistemas de exploración, por lo tanto la información a obtener es relacionada a cada uno de los candidatos presidenciales utilizando su nombre de usuario creado en Twitter, estos son:

- Lenín Moreno - @Lenin
- Guillermo Lasso - @LassoGuillermo

- Cynthia Viteri - @CynthiaViteri6
- Paco Moncayo - @PacoMoncayo
- Abdalá Bucaram - @daloes10
- Iván Espinel - @IvanEspinelM
- Washington Pezántez - @pesanteztwof
- Patricio Zuquilanda - @ZuquilandaDuque

Para recolectar los datos de una red social se utiliza la herramienta de Hadoop llamada **Flume**, la misma que según su sitio oficial (The Apache Software Foundation, 2017), es una herramienta altamente flexible en la obtención de grandes volúmenes de datos no estructurados y semi-estructurados de varias redes sociales de forma rápida, facilitando la adquisición de información personalizada por el usuario. En Twitter los datos son semi-estructurados (formato JSON), y al ser obtenidos a través de Flume se les da una estructura haciendo uso de los campos que posee un Tweet (fecha, id, texto, etc.).

Los campos seleccionados de la estructura del Tweet que fueron obtenidos a través de Flume, son almacenados en Hadoop utilizando la herramienta llamada **Hive**, la cual mediante una descripción rápida de su sitio oficial (The Apache Software Foundation, 2017) funciona como base de datos para el almacenamiento y procesamiento de la información obtenida de la red social Twitter.

En la figura 12 se muestra el flujo del caso de estudio 1:

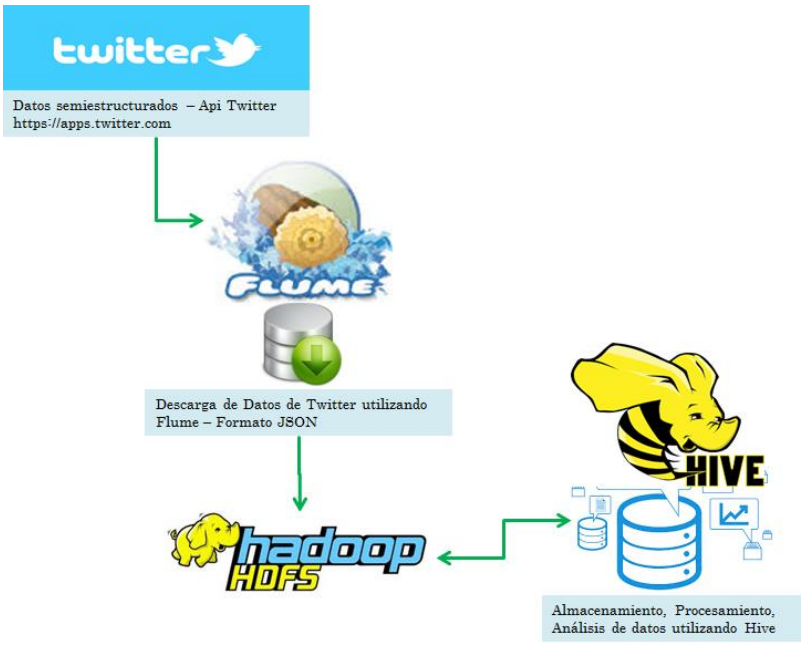


Figura 12: Flujo de caso de estudio 1
Fuente: Elaboración propia.

3.1.2. Caso de estudio 2

El segundo caso de estudio se centra en el procesamiento de datos generados en la segunda vuelta electoral 2017, la cual está almacenada en una base de datos relacional MySQL. Esta información es obtenida de la red social Twitter y está situada en tablas la misma que comparten identificadores relacionales entre sí.

La descarga de datos almacenados en MySQL hacia el HDFS de Hadoop se lo hace mediante la herramienta de Hadoop llamada **Sqoop,** la misma que según su sitio oficial (The Apache Software Foundation, 2017), actúa como una capa intermedia entre Hadoop y los sistemas de bases de datos relacionales.

Sqoop se utiliza de 3 maneras:

- Para importar datos de bases de datos relacionales (Teradata, Netezza, Oracle, MySQL, PostgreSQL, etc.) hacia Hadoop HDFS.
- Exportación de sistema de archivos Hadoop HDFS a bases de datos relacionales.
- Exportación de datos directamente entre las herramientas que forman parte del ecosistema de Hadoop (Hive, HBase, Cassandra, Pig, etc.).

Para el procesamiento y análisis de datos se hace uso de Apache **Pig** el cual según (The Apache Software Foundation, 2017) es utilizado para el análisis de grandes volúmenes de datos, manejando su propio lenguaje de alto nivel llamado Pig Latín que realiza en conjunto el análisis y evaluación de la información.

En la figura 13 se muestra el flujo del caso de estudio 2:

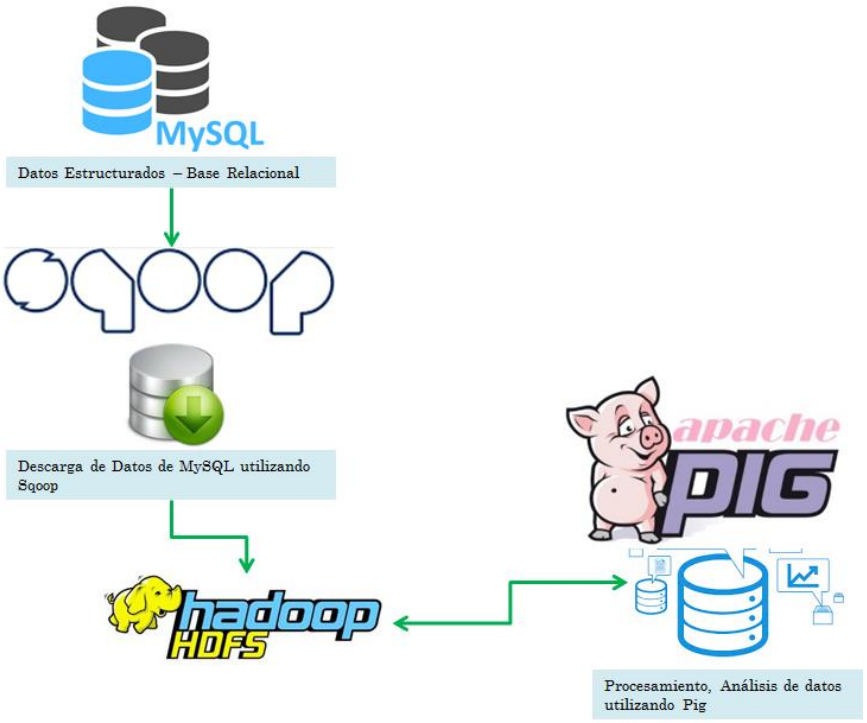


Figura 13: Flujo de caso de estudio 2
Fuente: Elaboración propia.

3.2. Arquitectura de las herramientas de Hadoop

3.2.1. Arquitectura de Flume

Flume es una herramienta distribuida del ecosistema de Hadoop que tiene como funcionalidad el recolectar, agregar y distribuir grandes cantidades de datos estructurados, semi-estructurados y no estructurados. La arquitectura que posee Flume es simple debido a que transmite los datos utilizando Streaming; esto facilita que el origen de datos sea configurable permitiendo que se pueda realizar el monitoreo de logs, descarga de información desde redes sociales (Facebook, Twitter, etc.) o también el tomar información desde correos electrónicos, entre otros. De igual manera el destino de los datos son configurables, permitiendo que Flume pueda funcionar sin el HDFS de Hadoop.

En la figura 14 se observa la arquitectura de Flume, la cual posee varios agentes que se encargan de recolectar la información de los generadores de datos (Facebook, Twitter). Posteriormente, otro agente llamado colector de datos toma los datos que son recogidos por los agentes y agrega la información en un almacén centralizados de datos tales como HDFS o HBase.

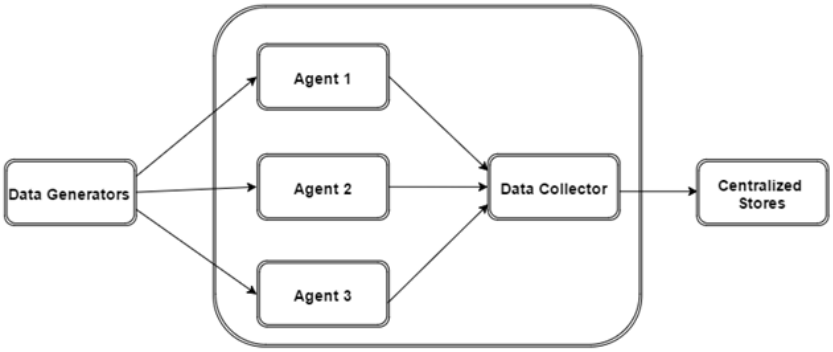


Figura 14: Arquitectura de Flume
Fuente: White T. (2015).

Cada agente representa un agente de canal de flujo el mismo que recibe los datos de los clientes u otros agentes y los reenvía a su próximo destino (Sink o disipador). En la figura 15 se muestra el diagrama de un agente Flume.

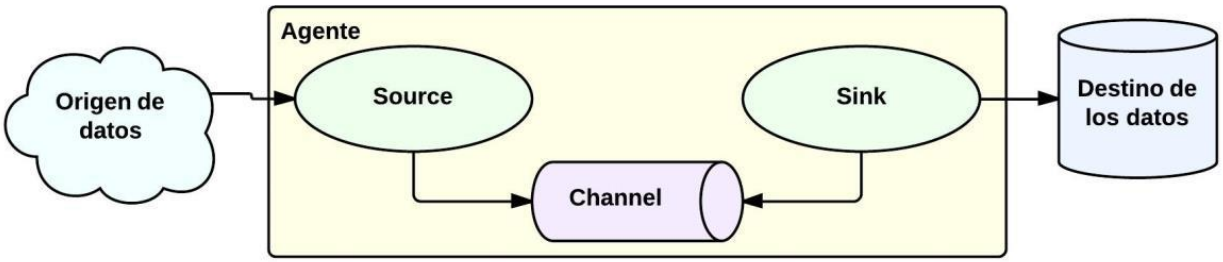


Figura 15: Agente Flume
Fuente: White T. (2015).

Un agente Flume está compuesto por 3 componentes:

- **Source (Fuente):** es el componente de un agente que se encarga de recibir los datos de los generadores de datos y los transporta a uno o más canales en forma de eventos Flume.

- **Channel (Canal):** se encarga de realizar un almacenamiento pasivo que recibe los eventos enviados por el source hasta que sea consumido por los Sink (disipador). El canal actúa como un puente entre el source y el sink.
- **Sink (Disipador):** se encarga de almacenar los datos en almacenes centralizados como HBase o HDFS. Consume los eventos generados por el canal y los entrega a su destino, pudiendo ser su destino otro agente o los almacenes de datos centrales.

De forma general, un agente Flume posee múltiples fuentes, canales y disipadores.

3.2.2. Arquitectura de Hive

Hive se ha convertido en un proyecto de Hadoop que proporciona maneras más fáciles para la creación, consulta y administración de grandes volúmenes de datos distribuidos que se encuentran contenidos en forma de tablas relacionales. Hive proporciona un tipo de lenguaje de consulta derivado del SQL el cual es llamado HiveQL o HQL. HiveQL está diseñado en base a MapReduce con el fin de aprovechar las características para el procesamiento de grandes cantidades de datos que se encuentren almacenados en Hadoop. El procesamiento y resultado de la información no se la dará en tiempo real.

En la figura 16 se observa la arquitectura que posee Hive según (Chen, 2015):

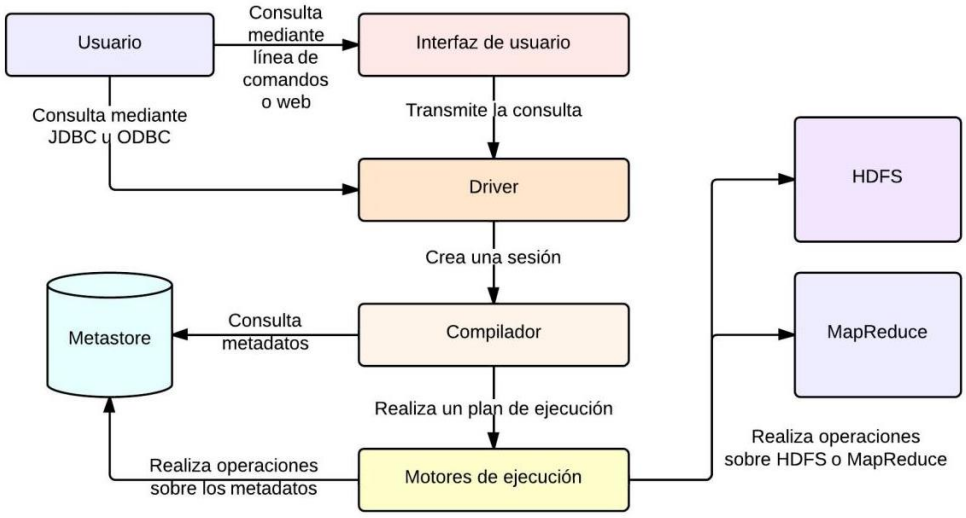


Figura 16: Arquitectura de Hive
Fuente: Chen C. (2015).

Los componentes de la arquitectura de Hive (Chen, 2015) se describen de esta manera:

- **Interfaz de usuario:** es el método que utiliza el usuario para el ingreso de las consultas. En Hive las interfaces de usuario son compatibles con interfaces Web, línea de comandos y Hive HD Insight (para Windows).
- **Driver:** recibe las consultas ingresadas en la interfaz de usuario y se encarga de implementar las sesiones. Adicional, recibe consultas vía interfaces JDBC y ODBC.
- **Compilador:** transforma y analiza la consulta (semánticamente y otras comprobaciones de lenguaje) para generar un plan de ejecución utilizando un metastore.
- **Metastore:** posee la información (metadatos) relacionada a la estructura de los datos que se encuentran dentro de Hive y que han sido compilados anteriormente. Estos metadatos tiene información relacionada al esquema o metadatos de tablas, bases de datos, las columnas de una tabla, sus tipos de datos y cartografía HDFS.
- **Motores de ejecución:** procesa y lleva a cabo el plan de ejecución que se ha planificado en el compilador. El motor de ejecución procesa la consulta, realiza operaciones sobre HDFS o MapReduce y genera los resultados.

El funcionamiento de Hive inicia con la consulta ingresada por el usuario, posteriormente el Driver toma la consulta crea una sesión y es enviada al compilador.

El compilador realiza la validación de la consulta y diseña un plan de ejecución. La consulta realizada por el compilador confirmará con el metastore los metadatos que sean necesarios para generar una división adecuada de la consulta que se deba realizar utilizando MapReduce. En algunos casos es necesario el uso de más de un MapReduce dependiente del tipo de consulta, esquema de datos o exigencia de la consulta. El plan diseñado por el compilador consta de varias fases que realiza un trabajo MapReduce, una operación sobre los metadatos y finaliza con una operación que la realiza sobre HDFS.

Los motores de ejecución toman y procesan el plan de ejecución, en el cual cada fase es ejecutada en un orden correcto. Cada uno de los resultados que se obtienen en cada una de las fases son almacenadas en ficheros temporales en el HDFS. Cada fichero es leído y se valida la consistencia de datos al momento de ejecutar una nueva fase. Al finalizar, se genera el resultado sobre un fichero HDFS final.

3.2.3. Arquitectura de Sqoop

Sqoop es una herramienta de Hadoop, la cual (Kumud, 2016) fue diseñada para transferir datos entre Hadoop y servidores de bases de datos relacionales. Sqoop actúa como una capa intermedia para importar y exportar los datos entre sistemas de bases de datos relacionales y

Hadoop, así como también directamente entre las herramientas que forman parte del ecosistema de Hadoop.

En la figura 17 se observa la arquitectura que posee Apache Sqoop según (White, 2015):

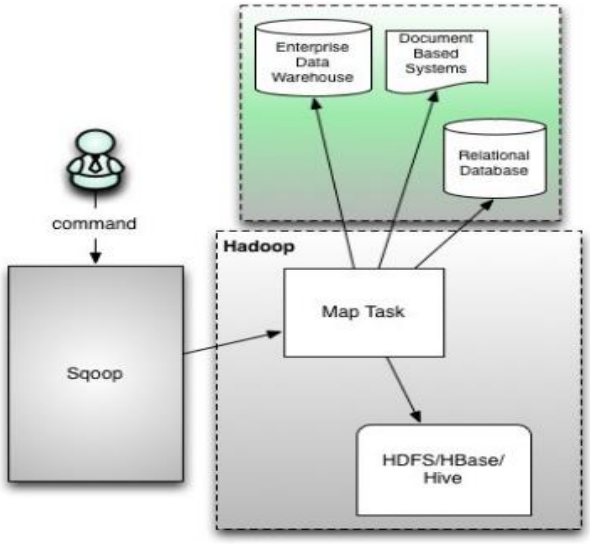


Figura 17: Arquitectura de Sqoop
Fuente: White T. (2015).

La arquitectura de funcionamiento de Sqoop (White, 2015) se describe de la siguiente manera:

- Sqoop proporciona una interfaz de línea de comandos para los usuarios finales, pero también se accede mediante las API de Java.
- Se analiza los argumentos proporcionados en la línea de comandos que es enviada por el usuario, si es correcta la sentencia Sqoop prepara el trabajo del Map para importar o exportar datos. Reduce no es utilizado dentro de Sqoop, ya que Reduce actúa cuando se realizan agregaciones. Sqoop realiza la importación/exportación de datos y no hace ninguna agregación.
- A continuación, la conexión especificada en la línea de comandos con la base de datos se lo realiza usando JDBC y va a buscar la parte de los datos asignados por Sqoop y lo escribe en HDFS, Hive, HBase, etc., dependiendo de si es importación o exportación.
- Para la importación se importan tablas individuales de sistemas de bases de datos relacionales a HDFS. Cada fila de una tabla es tratado como un registro en HDFS. Todos los registros se almacenan como datos en archivos de texto o como datos binarios.

- Para la exportación se exporta un conjunto de archivos de HDFS a sistemas de bases de datos relacionales. Los ficheros leídos por Sqoop contienen registros, que se asignan como filas en la tabla utilizando un delimitador especificado por el usuario.

De forma general, Sqoop automatiza la mayor parte del proceso de importación y exportación de datos, depende de la base de datos para describir el esquema de los datos a importar.

3.2.4. Arquitectura de Pig

Pig es una herramienta del ecosistema de Hadoop utilizada para el análisis de grandes volúmenes de datos, haciendo uso de un lenguaje de alto nivel que realiza en conjunto el análisis y evaluación de la información. Según (White, 2015) la principales funcionalidad de Pig es la de "crear una abstracción de lenguaje de procedimiento más simple sobre MapReduce para exponer una interfaz más parecida con Structured Query Language (SQL) para aplicaciones Hadoop". En base a esto, se ha creado Pig Latín el cual es un lenguaje de procesamiento de datos en alto nivel muy similar a la de SQL, el cual brinda una gran cantidad de tipos de datos y operadores que se utilizan para el procesamiento de la información. Un script de Pig Latín es automáticamente paralelizado y distribuido en un clúster de Hadoop, transformando un conjunto de datos en una serie de programas MapReduce.

En la figura 18 se observa la arquitectura que posee Apache Pig según (White, 2015):

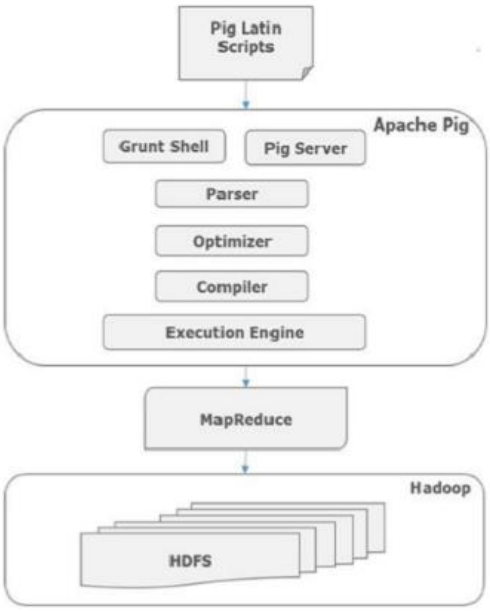


Figura 18: Arquitectura de Pig
Fuente: White T. (2015).

Los componentes de la arquitectura de Pig (White, 2015) se describen a continuación:

- **Analizador:** se encarga de comprobar la sintaxis de cada secuencia de comando en Pig, así como también la comprobación de tipos de datos y diversos controles aplicados a cada comando. El resultado del analizador es un DAG (en español: Gráfico Acíclico Dirigido) que contiene los estados de Pig Latín y sus operadores lógicos.
- **Optimizador:** el DAG que se generó en el Analizador, es pasado al optimizador de lógica para que se realicen varias transformaciones en el plan de ejecución y de esa manera obtener un DAG optimizado.
- **Compilador:** en este componente se compila el DAG optimizado convirtiéndolo en un plan de ejecución. En este plan se encuentran una serie de trabajos y pasos que se realiza utilizando el MapReduce de Hadoop.
- **Motor de ejecución:** El plan de ejecución obtenido en el Compilador, es ejecutado según las tareas del DAG. El motor de ejecución interactúa con el explorador de tareas de Hadoop para programar las tareas a ejecutarse en forma ordenada de acuerdo a los requisitos previos del DAG, y obteniendo de esta manera los resultados deseados.

3.3. Uso y funcionalidad de las herramientas de Hadoop

3.3.1. Funcionalidad y uso de Flume

3.3.1.1. Configuración de Flume.

Posterior a la instalación de Flume, es necesario configurarlo usando el archivo de configuración base, el mismo que contiene las propiedades Java con combinaciones pares de clave - valor.

Basándonos en la descripción de (White, 2015) el archivo de configuración de Flume debe poseer la siguiente información:

- Nombre de los componentes.
- Descripción de la fuente.
- Descripción del disipador (sink).
- Descripción del canal.
- Conectar el origen y el disipador para el canal.

Cada agente se diferencia de otro asignándoles un nombre único, ya que podemos poseer múltiples agentes en Flume.

Nombre de los componentes

Hace referencia a los nombres de las fuentes, disipador y los canales del agente.

```
agent_name.sources = source_name
agent_name.sinks = sink_name
agent_name.channels = channel_name
```

Figura 19: Nombre de los componentes Flume
Fuente: Elaboración propia.

El canal de flujo acepta diferentes fuentes, disipadores y canales. Para la transferencia de datos de Twitter, se usa un agente de identificación llamado TwitterAgent, como se muestra a continuación:

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
```

Figura 20: TwitterAgent en Flume
Fuente: Elaboración propia.

Descripción de la Fuente

Posee la descripción del tipo de fuente que se está utilizando. Adicional, se incluye las propiedades de la fuente que se va a utilizar.

```
agent_name.sources. source_name.type = value
agent_name.sources. source_name.property2 = value
agent_name.sources. source_name.property3 = value
```

Figura 21: Descripción de la fuente en Flume
Fuente: Elaboración propia.

Para Twitter la configuración de la descripción de la fuente de Flume es:

```
TwitterAgent.sources.Twitter.type = Twitter (type name)
TwitterAgent.sources.Twitter.consumerKey =
TwitterAgent.sources.Twitter.consumerSecret =
TwitterAgent.sources.Twitter.accessToken =
TwitterAgent.sources.Twitter.accessTokenSecret =
```

Figura 22: Descripción de la fuente con Twitter en Flume
Fuente: Elaboración propia.

Descripción del disipador (sink)

Recoge los datos desde el canal intermedio dentro de una transacción y los mueve a un repositorio externo, otra fuente o a un canal intermedio.

```
agent_name.sinks. sink_name.type = value
agent_name.sinks. sink_name.property2 = value
agent_name.sinks. sink_name.property3 = value
```

Figura 23: Descripción del disipador en Flume
Fuente: Elaboración propia.

Al utilizar una fuente Twitter la configuración de la descripción del disipador de Flume es:

```
TwitterAgent.sinks.HDFS.type = hdfs (type name)
TwitterAgent.sinks.HDFS.hdfs.path = HDFS directory's Path to store the data
```

Figura 24: Descripción del disipador con Twitter en Flume
Fuente: Elaboración propia.

Descripción del Canal

Funciona como almacén intermedio entre la fuente y el disipador (sink). La fuente es la encargada de escribir los datos en el canal y permanecen en él hasta que el disipador u otro canal los utilicen.

```
agent_name.channels.channel_name.type = value
agent_name.channels.channel_name. property2 = value
agent_name.channels.channel_name. property3 = value
```

Figura 25: Descripción del disipador con Twitter en Flume
Fuente: Elaboración propia.

En una fuente Twitter la configuración de la descripción del canal de Flume es:

```
TwitterAgent.channels.MemChannel.type = memory (type name)
```

Figura 26: Descripción del canal con Twitter en Flume
Fuente: Elaboración propia.

Conectar el origen y el disipador para el canal

A continuación, la configuración para las fuentes y los disipadores, se los une para establecer el canal.

```
agent_name.sources.source_name.channels = channel_name
agent_name.sinks.sink_name.channels = channel_name
```

Figura 27: Conectar el origen y el disipador para el canal en Flume
Fuente: Elaboración propia.

Para una fuente Twitter la configuración para enlazar las fuentes y los disipadores a un canal en Flume es:

```
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sinks.HDFS.channels = MemChannel
```

Figura 28: Conectar el origen y el disipador para el canal con Twitter en Flume
Fuente: Elaboración propia.

Ejecución de un agente en Flume

Para la ejecución de un agente Flume se debe iniciarlo con la siguiente sentencia:

```
$ bin/flume-ng agent --conf ./conf/ -f conf/twitter.conf Dflume.root.logger=DEBUG,console -n
TwitterAgent
```

Figura 29: Ejecución de un agente Flume
Fuente: Elaboración propia.

Dónde la sentencia tiene la siguiente información:

- **agent** es el comando para iniciar el agente de Flume.
- **--conf** o **-c<conf>** es archivo de configuración de uso en el directorio conf.
- **-f<file>** especifica una ruta de archivo de configuración (es opcional).
- **--name, -n <name>** hace referencia al nombre del agente.
- **-D property =value** establece un valor de propiedad del sistema Java.

3.3.1.2. Configuración de Flume para caso de estudio 1.

En el caso de estudio 1, se realiza la descarga de datos de Twitter en formato JSON mediante la configuración del agente Flume y posterior se activa la ejecución del agente almacenando la información descargada de Twitter en el HDFS de Hadoop.

Creación de una aplicación de Twitter

Para poder descargar la información desde el api de Twitter, se debe crear una aplicación de Twitter.

Como paso inicial, se debe ingresar al enlace https://apps.twitter.com/ con la cuenta de Twitter del usuario. Se abre una venta de Administración de Aplicaciones en la cual se crea, elimina y administra aplicaciones de Twitter.

A continuación, se procede a crear una nueva aplicación de Twitter y se ingresan los datos solicitados. La aplicación que he creado tiene como nombre “ObtenerTwitter”. Como referencia se tiene la figura 29.

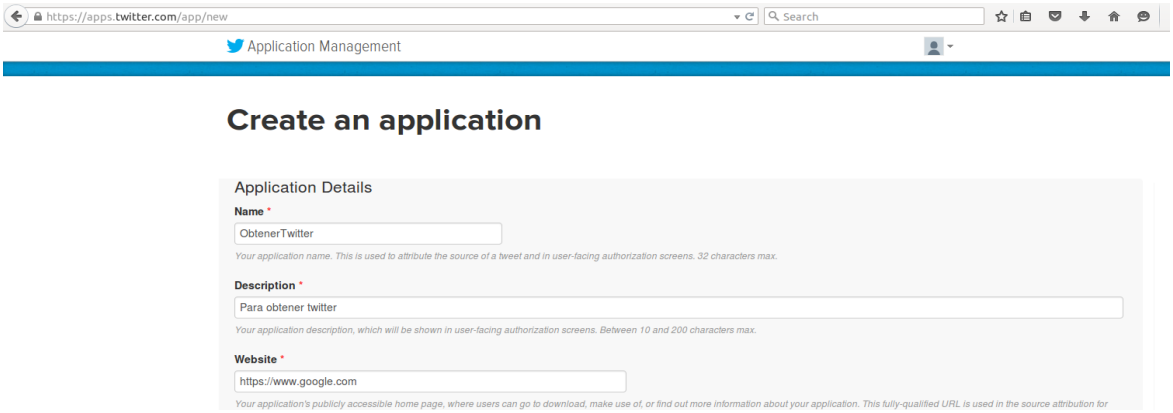


Figura 30: Creación de aplicación en Twitter
Fuente: Elaboración propia.

Posterior, se debe obtener los permisos de acceso de Twitter para descargar la información (keys y tokens). Estos se los obtiene en un botón llamado “Create my access token”. La información generada y que es necesaria para configurar el agente Flume se encuentra en los detalles de la aplicación creada, específicamente en la pestaña “Keys and Access Tokens”.

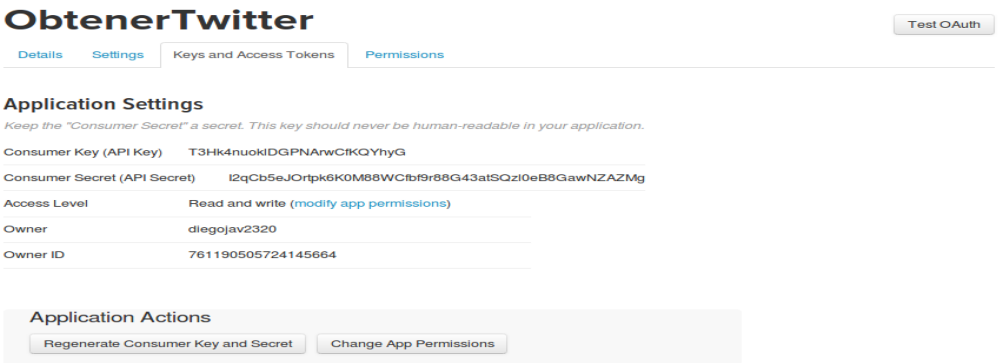


Figura 31: Obtener keys y tokens en Twitter
Fuente: Elaboración propia.

Los datos utilizados para el agente Flume son: Consumer Key, Consumer Secret, Access Token, y Access Token Secret.

Verificación de la instalación de Hadoop

Antes de configurar el agente Flume, es necesario que se encuentre instalado y levantado el servicio de Hadoop en la máquina virtual. Esto se explica en el ANEXO 3, en el apartado llamado Verificación de la instalación de Hadoop.

Configuración del agente Flume

Para la configuración del agente Flume es necesario crear un archivo de configuración en la carpeta "conf" del directorio de Flume, el cual es Home/work/apache-flume-1.6.0-bin/conf. En esta carpeta se ha procedido a crear el archivo de nombre "flume-twitter.conf" tal como se lo muestra en la siguiente figura:

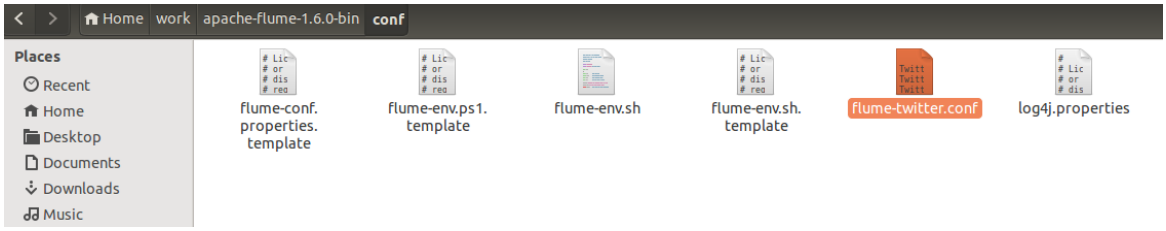


Figura 32: Creación de archivo de configuración en Flume
Fuente: Elaboración propia.

En el archivo creado "flume-twitter.conf" se incluye la siguiente información que es necesaria para el agente Flume de Twitter:

- **Componentes:** posee los nombres de las fuentes (source), canal (channel) y el disipador (sink) que hace relación a la configuración de Twitter.
- **Fuentes:** se establece el tipo y la ruta de descarga de los datos de Twitter. Adicional, se incluye criterios de búsqueda de información tales como palabras claves, coordenadas geográficas, etc.

  Para nuestro caso de estudio 1, se ha establecido que la información a descargar de Twitter sea en relación a palabras claves y que correspondan a cada uno de los candidatos presidenciales de la primera vuelta electoral 2017 utilizando su nombre de usuario creado en Twitter, estos son:

Tabla 2: Nombre de usuario candidatos presidenciales

Nombre Candidato	Nombre Usuario en Twitter
Lenín Moreno	@Lenin
Guillermo Lasso	@LassoGuillermo
Cynthia Viteri	@CynthiaViteri6
Paco Moncayo	@PacoMoncayo
Abdalá Bucaram	@daloes10
Iván Espinel	@IvanEspinelM
Washington Pezántez	@pesanteztwof
Patricio Zuquilanda	@ZuquilandaDuque

Fuente: Elaboración propia.

Otro filtro a aplicar al caso de estudio 1 es la descarga de tweets que son generados solo en el país Ecuador. Por lo cual, se ha tomado las coordenadas geográficas que limiten al territorio ecuatoriano.

- **Sink:** indica las características que son necesarias para almacenar la información en el HDFS de Hadoop. Incluye la ruta de destino, canal, el tipo de formato de los archivos que se van a almacenar, tamaño del archivo, entre otras características.

  En la ruta de destino se tiene el path del HDFS en donde se almacena la información descargada de Twitter. Para nuestro caso de estudio 1 se crearon varias carpetas de descargas en diferentes fechas de la primera vuelta electoral 2017. Los archivos JSON de Twitter se almacenaron con el nombre de “twitter + número_ascendente”.
- **Canal:** En la sección del canal se establece la configuración que corresponde al tipo de memoria y la capacidad que tiene el canal.

La configuración aplicada al archivo “flume-twitter.conf” de la máquina virtual se lo observa en la figura 32:

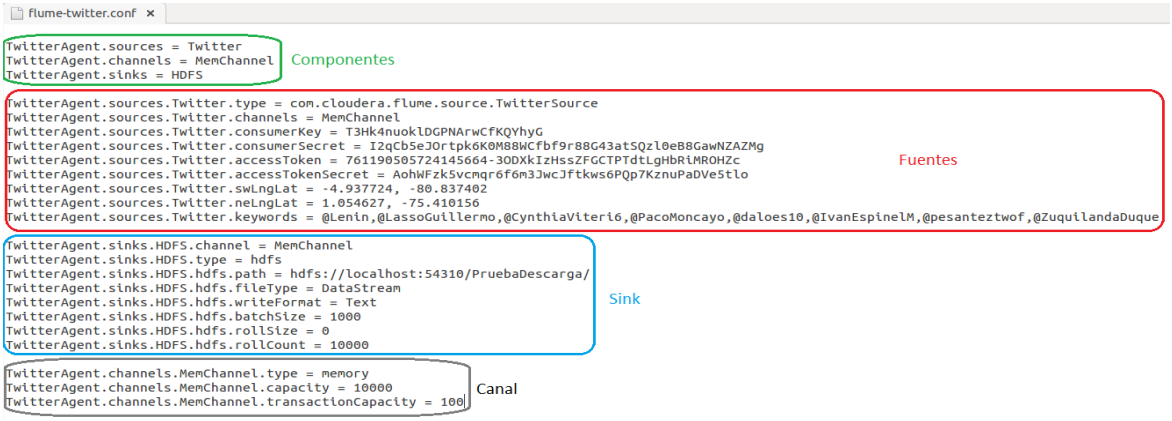


Figura 33: Archivo de configuración final de Flume
Fuente: Elaboración propia.

Ejecución del agente Flume

Al terminar la configuración del agente Flume, es necesario ejecutarlo para que se inicie la descarga de los datos de Twitter y almacenarlos en los archivos del HDFS de Hadoop.

Previamente, se debe direccionar la terminal de la máquina virtual a la carpeta en la que se encuentra Flume. Para acceder al directorio de Flume se debe usar el siguiente comando sobre la terminal de Linux:

```
$ cd $FLUME_HOME
```

Figura 34: Abrir directorio Flume

Fuente: Elaboración propia.

La descarga de los datos de Twitter se lo hace mediante la ejecución del siguiente comando:

```
$ bin/flume-ng agent -n TwitterAgent --conf ./conf/ -f conf/flume-twitter.conf -
Dflume.root.logger=DEBUG, console
```

Figura 35: Ejecución del agente Flume de Twitter
Fuente: Elaboración propia.

Al iniciarse la descarga, en la terminal del sistema se muestra el proceso de búsqueda de los tweets según los filtros que se hayan parametrizado en la configuración del agente Flume.

Figura 36: Descarga y almacenamiento del agente Flume
Fuente: Elaboración propia.

Almacenamiento de tweets en el HDFS

El almacenamiento de los tweets que descarga el agente Flume, se lo observa accediendo al HDFS de Hadoop. El puerto predeterminado para acceder a Hadoop es el 50070 y se debe utilizar la url http://localhost:50070/ para obtener los servicios de Hadoop en el navegador.

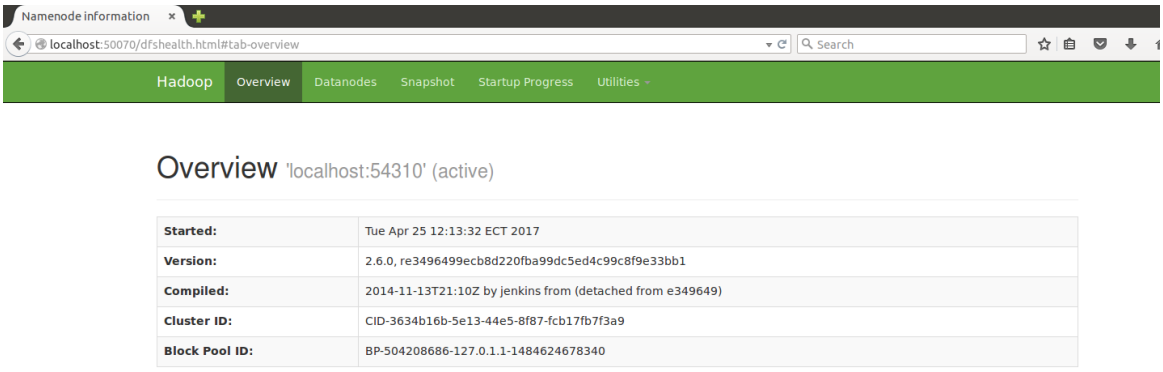


Figura 37: HDFS de Hadoop
Fuente: Elaboración propia.

En el menú “Utilities” se debe seleccionar “Browse the file system” para desplegar las carpetas que contiene HDFS, entre los que se encuentran las carpetas contenedoras de los tweets.

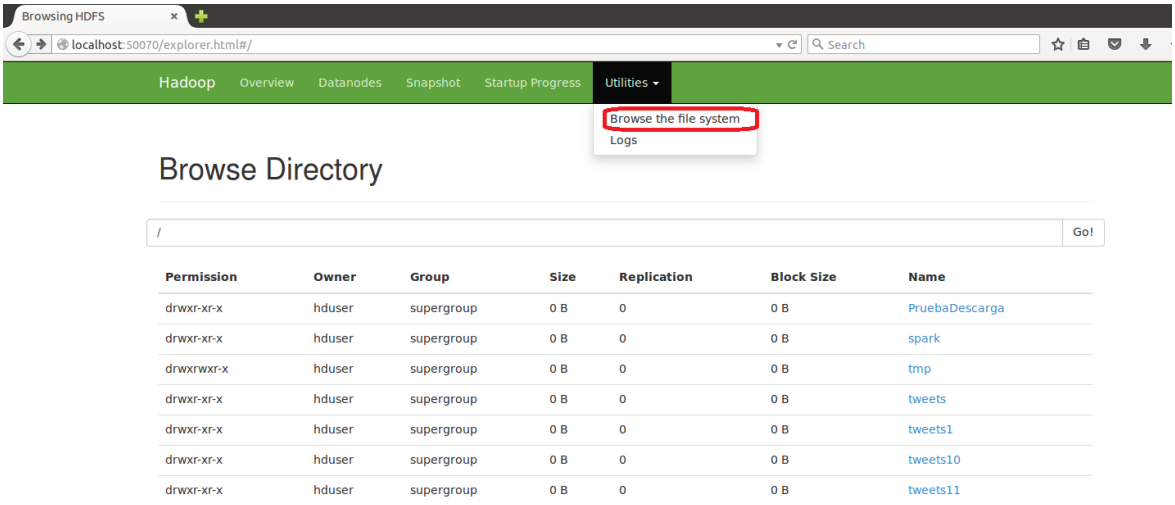


Figura 38: Carpetas del HDFS
Fuente: Elaboración propia.

Los datos descargados en HDFS para el caso de estudio 1 se encuentran dentro de las carpetas “twitter + número_ascendente”. Al ingresar a alguna de estas carpetas se visualiza los archivos generados con el nombre de “FlumeData”. Estos archivos contienen todos los tweets descargados en formato JSON.

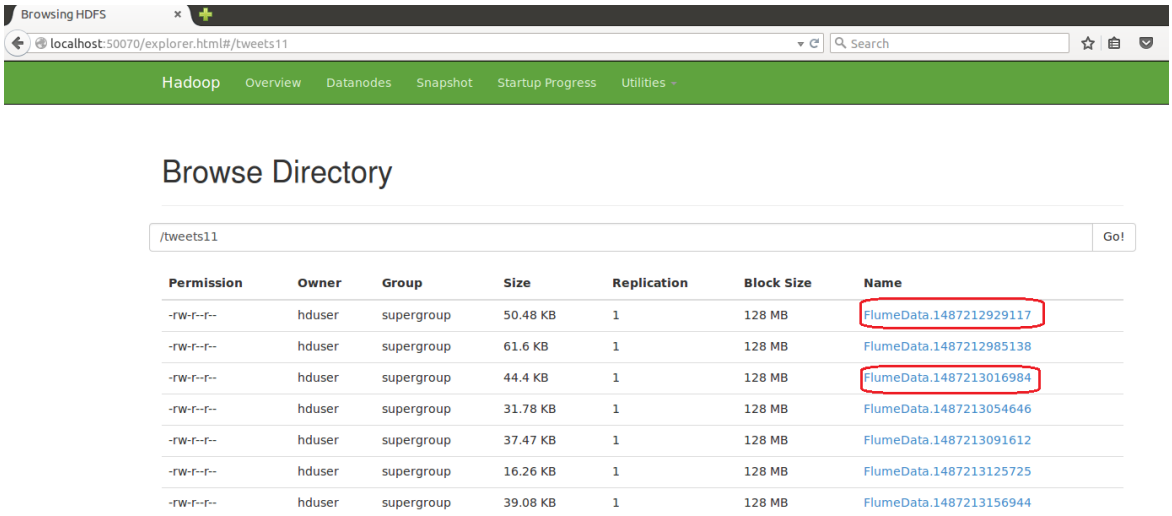


Figura 39: FlumeData
Fuente: Elaboración propia.

3.3.2. Funcionalidad y uso de Hive

3.3.2.1. Comandos básicos y sintaxis para Hive.

Los usuarios que han utilizado el lenguaje SQL están familiarizados con los comandos DDL (Data Definition Language, en español Lenguaje de Definición de Datos) los cuales son utilizados para definir y cambiar la estructura de una base de datos o tabla en Hive.

Tabla 3: Comandos DDL

Comandos DDL	Descripción
CREATE	Creación de base de datos y tablas
DROP	Eliminar base de datos y tablas
TRUNCATE	Truncar tabla
ALTER	Alterar base de datos y tablas
SHOW	Mostrar base de datos, tablas, propiedades de tablas, particiones, funciones e índices
DESCRIBE	Describir base de datos, tablas y vistas

Fuente: Elaboración propia.

Según (White, 2015) los comandos básicos y sintaxis para Hive son:

Create en base de datos Hive

Se utiliza para crear bases de datos en Hive (los valores mencionados entre corchetes [] son opcionales).

```
CREATE (DATABASE) [IF NOT EXISTS] database_name
  [COMMENT database_comment]
  [LOCATION hdfs_path]
  [WITH DBPROPERTIES (property_name=property_value, ...)];
```

Figura 40: Create en base de datos Hive
Fuente: Elaboración propia.

Drop en base de datos Hive

Se utiliza para eliminar una base de datos ya creada en Hive.

```
DROP (DATABASE) [IF EXISTS] database_name [RESTRICT|CASCADE];
```

Figura 41: Drop en base de datos Hive
Fuente: Elaboración propia.

En la sintaxis del comando DROP sobre la base de datos, se utiliza la cláusula "if exists" para evitar cualquier error que pueda ocurrir si el programador intenta eliminar una base de datos que no existe.

Alter en base de datos Hive

Se utiliza para cambiar los metadatos de cualquiera de las bases de datos de Hive.

```
ALTER (DATABASE) database_name SET DBPROPERTIES (property_name=property_value, ...);
```

Figura 42: Alter en base de datos Hive
Fuente: Elaboración propia.

Show en base de datos Hive

Se utiliza para ver la lista de bases de datos y tablas existentes en el esquema actual.

```
SHOW databases;
```

Figura 43: Show en base de datos Hive
Fuente: Elaboración propia.

Use en base de datos Hive

Se utiliza para seleccionar una base de datos específica para la sesión en la que se ejecuta las consultas de Hive.

```
SHOW (DATABASE) database_name;
```

Figura 44: Use en base de datos Hive
Fuente: Elaboración propia.

Create de tabla en Hive

Se utiliza para crear una tabla en la base de datos existente que se está utilizando.

```
CREATE TABLE [IF NOT EXISTS] [db_name.]table_name    --
  [(col_name data_type [COMMENT col_comment], ...)]
  [COMMENT table_comment]
  [LOCATION hdfs_path]
```

Figura 45: Create de tabla en Hive
Fuente: Elaboración propia.

Hay 2 tipos de tablas en Hive

- **Interna:** es una tabla normal de la base de datos donde se almacenan y consultar los datos de Hive. Al eliminar estas tablas los datos almacenados en ellos también se eliminan definitivamente.

```
CREATE TABLE [IF NOT EXISTS] [db_name.]table_name
```

Figura 46: Create de tabla interna en Hive
Fuente: Elaboración propia.

- **Externa:** se almacenan los archivos en el servidor HDFS, pero las tablas no están vinculadas al archivo de origen por completo. Si se elimina una tabla externa, el archivo permanece en el servidor HDFS.

```
CREATE EXTERNAL TABLE [IF NOT EXISTS] [db_name.]table_name
```

Figura 47: Create de tabla externa en Hive
Fuente: Elaboración propia.

Creación de una tabla en Hive copiando un esquema de tabla existente

Hive permite al usuario crear una nueva tabla mediante la replicación del esquema de una tabla existente. La réplica aplica solo al esquema, pero no los datos. Al crear la nueva tabla, se especifica el parámetro de ubicación.

```
CREATE TABLE [IF NOT EXISTS] [db_name.]table_name   Like [db_name].existing_table
[LOCATION hdfs_path]
```

Figura 48: Creación de una tabla en Hive copiando un esquema de tabla existente
Fuente: Elaboración propia.

Drop de tabla en Hive

Elimina la tabla y todos los datos asociados a ella en el metastore de Hive.

```
DROP TABLE [IF EXISTS] table_name [PURGE];
```

Figura 49: Drop de tabla en Hive
Fuente: Elaboración propia.

Show de tabla en Hive

Se utiliza para ver la lista de tablas existentes en el esquema de la base de datos actual.

```
SHOW (TABLE) table_name
```

Figura 50: Show de tabla en Hive
Fuente: Elaboración propia.

Comandos DML (Data Manipulation Language) en Hive

En Hive los comandos DML (Data Manipulation Language, en español Lenguaje de Manipulación de Datos) se utilizan para insertar y consultar los datos de las tablas una vez que se haya definido la estructura y la arquitectura de la base de datos mediante los comandos DDL mencionados anteriormente.

Cargar los datos en tabla de Hive

Los datos se cargan de 2 formas en Hive, ya sea desde un archivo local o desde HDFS a Hive.

Para cargar los datos desde un archivo local se debe usar el comando:

```
hadoop fs –copyFromLocal /home/user/(ubicacion_archivo) hdfs://localhost:50070/(nueva_ubicacion)
```

Figura 51: Cargar los datos en la tabla Hive desde archivo local
Fuente: Elaboración propia.

Otra forma de cargar datos es cargarlo desde HDFS a Hive utilizando el comando LOAD (los valores mencionados entre corchetes [] son opcionales).

```
LOAD DATA [LOCAL] INPATH 'hdfsfilepath/localfilepath' [OVERWRITE] INTO TABLE
existing_table_name
```

Figura 52: Cargar los datos en la tabla Hive con LOAD
Fuente: Elaboración propia.

Si no se especifica la palabra clave LOCAL, Hive necesitará el directorio completo de la ubicación de archivo. Sin embargo, si se especifica LOCAL entonces asume las siguientes condiciones:

- Se asumirá que es una ruta HDFS y busca el archivo en HDFS.
- Si la ruta no es absoluta, entonces Hive trata de localizar el archivo en el directorio /user/ en HDFS.

El uso de la palabra clave OVERWRITE durante la importación significa que los datos se sobrescribirán, es decir, borrará los datos antiguos y pondrá nuevos datos, de lo contrario sólo añadiría los nuevos datos.

Select en tablas de Hive

Se utiliza para recuperar los datos de una tabla.

```
SELECT * FROM existing_table
```

Figura 53: Select en tablas de Hive
Fuente: Elaboración propia.

Where en tablas de Hive

Filtra los datos de una tabla dando un resultado finito. Los operadores y funciones generan una expresión que satisface la condición.

```
SELECT * FROM existing_table WHERE (condicion)
```

Figura 54: Where en tablas de Hive
Fuente: Elaboración propia.

Union en tablas de Hive

Se utiliza para combinar el resultado de varias sentencias SELECT en un conjunto de resultados único.

```
SELECT [campos| *] from existing_table
UNION [ALL | DISTINCT]
SELECT [campos| *] from existing_table
UNION [ALL | DISTINCT]
SELECT [campos| *] from existing_table
```

Figura 55: Union en tablas de Hive
Fuente: Elaboración propia.

Count en tablas de Hive

Se utiliza para contar el número de filas de una tabla.

```
SELECT COUNT ([campos|*]) from existing_table
```

Figura 56: Count en tablas de Hive
Fuente: Elaboración propia.

3.3.2.2. Aplicación de los comandos y sintaxis de Hive en caso de estudio 1.

Verificación de la instalación de Hadoop

Antes de utilizar Hive, es necesario que se encuentre instalado y levantado el servicio de Hadoop en la máquina virtual. Esto se explica en el ANEXO 3 en el apartado llamado Verificación de la instalación de Hadoop.

Inicio de Hive

Posterior al inicio de Hadoop, se inicia Hive en la máquina virtual ejecutando el siguiente comando en la terminal del sistema:

```
$ hive
```

Figura 57: Inicio de Hive
Fuente: Elaboración propia.

Exportar datos de HDFS a Hive

El proceso de manejo de la información de Twitter se lo hace en varios pasos en Hive: creación de la base de datos, creación de una tabla, exportar datos de HDFS a la tabla creada, ejecución de los scripts de consulta sobre Hive, obtención de resultados y datos.

Creación de la base de datos

Para crear la base de datos en Hive, lo haremos usando el comando CREATE DATABASE con el nombre de la base de datos “base_Hive”.

```
hive> create database if not exists base_hive;
```

Figura 58: Creación de base de datos en Hive
Fuente: Elaboración propia.

Para listar las bases de datos existentes en Hive se utiliza el siguiente comando:

```
hive> show databases;
```

Figura 59: Mostrar base de datos en Hive
Fuente: Elaboración propia.

Luego de la creación de la base de datos “base_Hive” en Hive, se va a asignar su uso con el comando USE ejecutándola sobre la terminal del sistema:

```
hive> use base_hive;
```

Figura 60: Usar base de datos en Hive
Fuente: Elaboración propia.

Creación de la tabla

Previo al paso de la data de HDFS a la tabla nueva que se creará, es necesario descargar el archivo hive-serdes-1.0-SNAPSHOT.jar de la url http://files.cloudera.com/samples/hive-serdes-1.0-SNAPSHOT.jar. Este archivo de extensión .jar es una librería de java que se encarga de realizar el mapeo de los archivos en formato JSON que se han descargado de Twitter y que nos sirve para migrar los datos en la tabla nueva de Hive.

Luego de la descarga del archivo hive-serdes-1.0-SNAPSHOT.jar hay que registrarlo en Hive, esto se lo hace con el comando ADD y con el directorio de ubicación del archivo en la máquina virtual.

```
hive> add jar /home/diego/Downloads/hive-serdes-1.0-SNAPSHOT.jar;
```

Figura 61: ADD de librería .jar en Hive
Fuente: Elaboración propia.

La data de HDFS es migrada a varias tablas externas. Los campos a tomar de los tweets y que se encuentra en formato JSON son:

- **id:** identificador único del tweet.
- **created_at:** fecha de creación del tweet.
- **text:** texto publicado del tweet y escrito por el usuario.
- **retweet_count:** número de veces que es retweeteado el tweet.

Las tablas externas que se ha creado utilizan el nombre "twitter + (número carpeta HDFS)_data". El "número de carpeta HDFS" es la carpeta en la que se descarga la data de twitter utilizando la herramienta FLUME.

Para la creación de la tabla en la que se descarga la data de twitter, se lo realiza leyendo cada registro con la librería JSONSerDe que previamente fue agregado en Hive. La tabla se llenará con los datos del archivo JSON que coincidan con los campos especificados en la tabla. En la terminal del sistema se ejecuta el siguiente comando de creación:

```
hive> CREATE EXTERNAL TABLE if not exists tweets11_data(
    >  id BIGINT,
    > created_at STRING,
    > text STRING ,
    > retweet_count  INT )
    > ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
    > LOCATION '/tweets11';
```

Figura 62: Creación de tabla externa en Hive
Fuente: Elaboración propia.

Luego de crear tablas externas, se crea tablas internas para que la información de estas tablas se almacene en un espacio de HDFS. El paso de información de las tablas externas a una interna se lo debe hacer en dos partes: crear la tabla interna y llenar la tabla interna.

Para crear la tabla interna con el nombre "data_primera_vuelta" y que tome la estructura de la tabla externa se ejecuta el siguiente comando en Hive:

```
hive> create table data_primera_vuelta as select * from tweets11_data;
```

Figura 63: Creación de tabla interna en Hive
Fuente: Elaboración propia.

Para continuar llenando de datos la tabla “data_primera_vuelta” con los datos de las tablas externas se ejecutará el siguiente comando en Hive:

```
hive> insert into data_primera_vuelta select * from tweets12_data;
```

Figura 64: Llenado de tabla interna en Hive
Fuente: Elaboración propia.

Manipulación de Big Data en Hive

Al tener la información en una tabla de Hive llamada “data_primera_vuelta” se procede a ejecutar las consultas que nos permiten darnos información de tendencias de la primera vuelta electoral de acuerdo a las sentencias de consulta que se quiera realizar.

Se ha realizado la búsqueda de información en 5 consultas, las cuales han arrojado los siguientes datos:

- **Total de registros en 1ra vuelta**

```
hive> select count(*) from data_primera_vuelta;
```

Figura 65: Sentencia para total de registros en 1ra vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

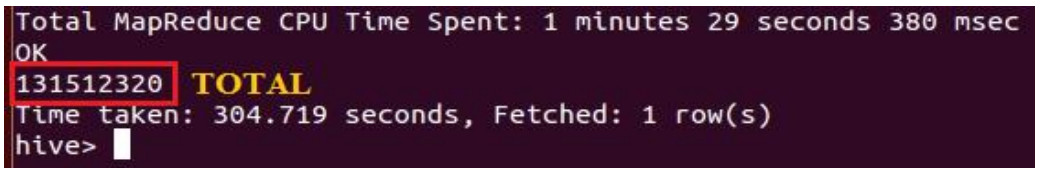


Figura 66: Resultado del total de registros en 1ra vuelta
Fuente: Elaboración propia.

- **Número de veces que se mencionaron a los candidatos en 1ra vuelta**

```
hive> select '@Lenin fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@Lenin%'
> union
> select '@CynthiaViteri6 fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@CynthiaViteri6%'
> union
> select '@daloes10 fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@daloes10%'
> union
> select '@ZuquilandaDuque fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@ZuquilandaDuque%'
> union
> select '@pesanteztwof fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@pesanteztwof%'
> union
> select '@LassoGuillermo fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@LassoGuillermo%'
> union
> select '@PacoMoncayo fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@PacoMoncayo%'
> union
> select '@IvanEspinelM fue mencionado' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%@IvanEspinelM%';
```

Figura 67: Sentencia para número de veces que se mencionaron a los candidatos en 1ra vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

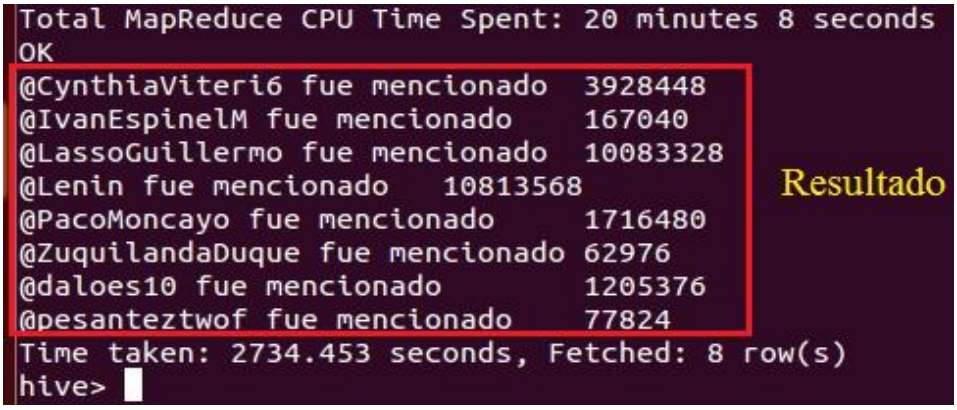


Figura 68: Resultado número de veces que se mencionaron a los candidatos en 1ra vuelta
Fuente: Elaboración propia.

- **Cantidad de tweets que publicó cada candidato en 1ra vuelta**

```
hive> select 'Cantidad de tweets de @Lenin' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @Lenin%'
> union
> select 'Cantidad de tweets de @CynthiaViteri6' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @CynthiaViteri6%'
> union
> select 'Cantidad de tweets de @daloes10' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @daloes10%'
> union
> select 'Cantidad de tweets de @ZuquilandaDuque' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @ZuquilandaDuque%'
> union
> select 'Cantidad de tweets de @pesanteztwof' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @pesanteztwof%'
> union
> select 'Cantidad de tweets de @LassoGuillermo' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @LassoGuillermo%'
> union
> select 'Cantidad de tweets de @PacoMoncayo' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @PacoMoncayo%'
> union
> select 'Cantidad de tweets de @IvanEspinelM' as nombre, count(id) as cantidad from data_primera_vuelta where text like '%RT @IvanEspinelM%';
```

Figura 69: Sentencia para cantidad de tweets que publicó cada candidato en 1ra vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

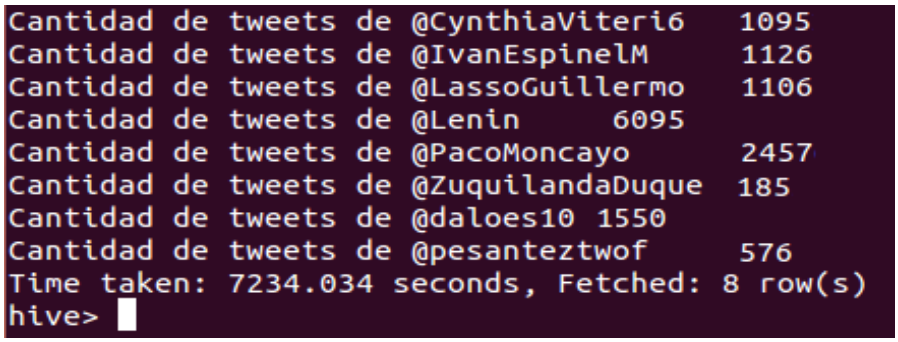


Figura 70: Resultado del número de tweets que publicó cada candidato en 1ra vuelta
Fuente: Elaboración propia.

- **Tweets más retweeteados (texto y mayor a 200000 veces) en 1ra vuelta**

```
hive> select 'Texto:' as texto, text, 'Cantidad de veces:' as cantidad, count(text) from
data_primera_vuelta group by text having count(text) > 200000 order by text;
```

Figura 71: Sentencia para tweets más retweeteados en 1ra vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

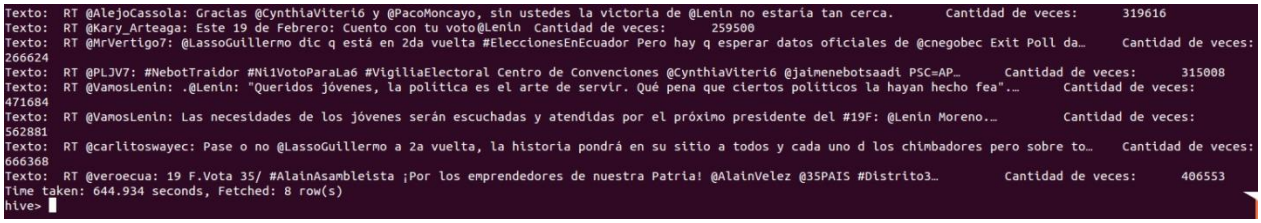


Figura 72: Resultado del número de tweets más retweeteados en 1ra vuelta
Fuente: Elaboración propia.

- **Cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta**

```
hive> select 'Cantidad de tweets de Rafael Correa hablando de @Lenin:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@Lenin%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @CynthiaViteri6:' as nombre, count(id) as cantidad from
data_primera_vuelta where  text like '%RT @MashiRafael:%' and text like '%@CynthiaViteri6%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @daloes10:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@daloes10%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @ZuquilandaDuque:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@ZuquilandaDuque%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @pesanteztwof:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@pesanteztwof%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @LassoGuillermo:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@LassoGuillermo%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @PacoMoncayo:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@PacoMoncayo%'
> union
> select 'Cantidad de tweets de Rafael Correa hablando de @IvanEspinelM:' as nombre, count(id) as cantidad from
data_primera_vuelta where text like '%RT @MashiRafael:%' and text like '%@IvanEspinelM%';
```

Figura 73: Sentencia para número de tweets de Rafael Correa por candidato en 1ra vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

```
Cantidad @MashiRafael hablando de @CynthiaViteri6:      88
Cantidad @MashiRafael hablando de @IvanEspinelM:        28
Cantidad @MashiRafael hablando de @LassoGuillermo:      120
Cantidad @MashiRafael hablando de @Lenin:       101
Cantidad @MashiRafael hablando de @PacoMoncayo: 51
Cantidad @MashiRafael hablando de @ZuquilandaDuque:     12
Cantidad @MashiRafael hablando de @daloes10:    108
Cantidad @MashiRafael hablando de @pesanteztwof:        12
Time taken: 3233.652 seconds, Fetched: 8 row(s)
hive>
```

Figura 74: Resultado del número de tweets de Rafael Correa por candidato en 1ra vuelta
Fuente: Elaboración propia.

3.3.3. Funcionalidad y uso de Sqoop

3.3.3.1. Comandos básicos y sintaxis para Sqoop.

Según (White, 2015) los comandos básicos y sintaxis para Sqoop son:

Sqoop-Import

Permite la importación de una tabla desde una base de datos relacional hacia el HDFS de Hadoop.

Sintaxis genérica:

```
$ sqoop import (generic args) (import args)
$ sqoop-import (generic args) (import args)
```

Figura 75: Sqoop importación
Fuente: Elaboración propia.

Importación de una tabla en HDFS

Sintaxis:

```
$ sqoop import --connect --table --username --password --target-dir
```

Figura 76: Importación de una tabla en HDFS
Fuente: Elaboración propia.

Tabla 4: Descripción de importación de una tabla en HDFS

Parámetro	Descripción
--connect	Asigna el URL de JDBC y se conecta a la base de datos
--table	Nombre de tabla de origen a importar
--username	Nombre de usuario para conectarse a la base de datos
--password	Contraseña del usuario de conexión
--target-dir	Importa los datos al directorio especificado

Fuente: Elaboración propia.

Importación de datos seleccionados de una tabla

Sintaxis:

```
$ sqoop import --connect --table --username --password --columns --where
```

Figura 77: Importación de datos seleccionados de una tabla
Fuente: Elaboración propia.

Tabla 5: Descripción de importación de datos seleccionados de una tabla

Parámetro	Descripción
--columns	Selecciona el subconjunto de columnas de la tabla señalada
--where	Recupera los datos que satisfacen la condición

Fuente: Elaboración propia.

Notas:

- En la cadena de conexión JDBC, el host de la base de datos no debe usarse como "localhost", ya que Sqoop lanza asignadores en múltiples nodos de datos y el asignador no puede conectarse al host de la base de datos.
- El parámetro --password es inseguro ya que se lo lee desde la línea de comandos. La opción –P es recomendable utilizar, ya que solicita la contraseña en la consola. De lo contrario, se recomienda utilizar el parámetro --password-file apuntando al archivo que contiene la contraseña (es necesario asegurarse de que se ha revocado el permiso a usuarios no autorizados).

Argumentos útiles en la importación de Sqoop:

Tabla 6: Descripción de argumentos útiles en la importación de Sqoop

Argumento	Descripción
--num-mappers, -m	Número de mapeadores que se lanzarán
--fields-terminated-by	Separador de campos
--lines-terminated-by	Separador de fin de línea

Fuente: Elaboración propia.

Sqoop-Export

Exporta un conjunto de archivos de un directorio HDFS a las tablas de una base de datos. La tabla de destino debe existir en la base de datos relacional.

Sintaxis genérica:

```
$ sqoop export (generic args) (export args)
$ sqoop-export (generic args) (export args)
```

Figura 78: Sqoop-Export sintaxis genérica
Fuente: Elaboración propia.

El comando de exportación de Sqoop prepara instrucciones INSERT con un conjunto de datos de entrada y luego los inserta en la base de datos relacional. Se realizan validaciones de clave primaria antes de realizar el insert.

Sintaxis:

```
$ sqoop-export ---connect --username --password --export-dir
```

Figura 79: Sqoop-Export
Fuente: Elaboración propia.

Sqoop-List-Database

Se utiliza para listar todas las bases de datos disponibles en el servidor de la base de datos relacional.

Sintaxis genérica:

```
$ sqoop list-databases (generic args) (list databases args)
$ sqoop-list-databases (generic args) (list databases args)
```

Figura 80: Sqoop-List-Database sintaxis genérica
Fuente: Elaboración propia.

Sintaxis:

```
$ sqoop list-databases --connect
```

Figura 81: Sqoop-List-Database
Fuente: Elaboración propia.

Sqoop-List-Tables

Se utiliza para listar todas las tablas en una base de datos especificada.

Sintaxis genérica:

```
$ sqoop list-tables (generic args) (list tables args)
$ sqoop-list-tables (generic args) (list tables args)
```

Figura 82: Sqoop-List-Tables sintaxis genérica
Fuente: Elaboración propia.

Sintaxis:

```
$ sqoop list-tables –connect
```

Figura 83: Sqoop-List-Tables
Fuente: Elaboración propia.

3.3.3.2. Aplicación de los comandos y sintaxis de Sqoop en caso de estudio 2.

El caso de estudio 2 está centrado en la manipulación de datos almacenados en una base de datos relacional MySQL que se encuentra instalada en la máquina virtual. La información almacenada en MySQL es obtenida de la red social Twitter la cual que es tratada y procesada para situar los datos en varias tablas relacionales y que comparten identificadores que las relacionan entre sí.

Para facilitar el uso de Sqoop se ha consolidado en MySQL una sola base de datos llamada “base_produccion” y en una sola tabla llamada “data_consolidada_2v” todos los datos de Twitter que se encuentran en varias tablas relacionales.

Verificación de la instalación de Hadoop

Antes de utilizar Hive, es necesario que se encuentre instalado y levantado el servicio de Hadoop en la máquina virtual. Esto se explica en el ANEXO 3 en el apartado llamado Verificación de la instalación de Hadoop.

Listar las bases de datos de MySQL

```
$ sqoop list-databases --connect jdbc:mysql://localhost:3306 --username root --password password123
```

Figura 84: Sentencia para listar las bases de datos de MySQL
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

Figura 85: Resultado de listar las bases de datos de MySQL
Fuente: Elaboración propia.

Listar las tablas de una base de datos de MySQL

```
$ sqoop list-tables --connect jdbc:mysql://localhost:3306/base_produccion --username root --password password123
```

Figura 86: Sentencia para listar las tablas de una base de datos de MySQL
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

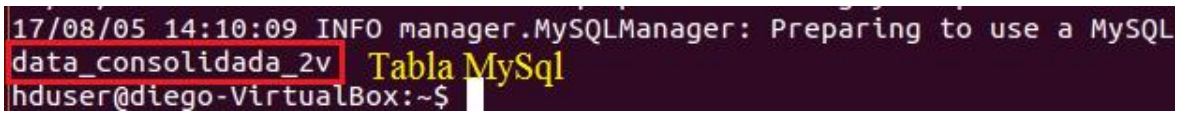


Figura 87: Resultado de listar las tablas de una base de datos de MySQL
Fuente: Elaboración propia.

Descarga de datos de MySQL a HDFS

```
sqoop import --connect jdbc:mysql://localhost:3306/base_produccion --username root --password password123 --table data_consolidada_2v --fields-terminated-by '|' --lines-terminated-by '\n' --target-dir /dataMySql;
```

Figura 88: Sentencia para descargar datos de MySQL a HDFS
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

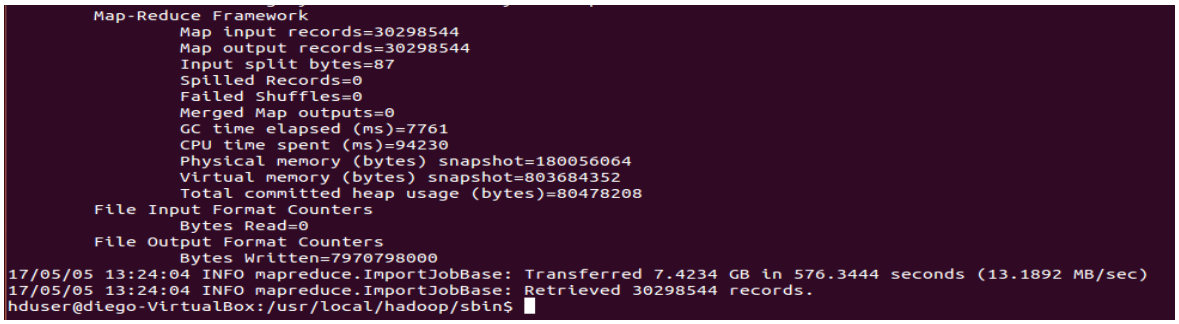


Figura 89: Resultado de descargar datos de MySQL a HDFS
Fuente: Elaboración propia.

Los datos descargados de MySQL son almacenados en el directorio HDFS que es especificado en la sentencia de Sqoop. Para nuestro caso de estudio 2 la información de MySQL es almacenada en la carpeta de nombre “dataMySql” como lo indican la siguiente figura:

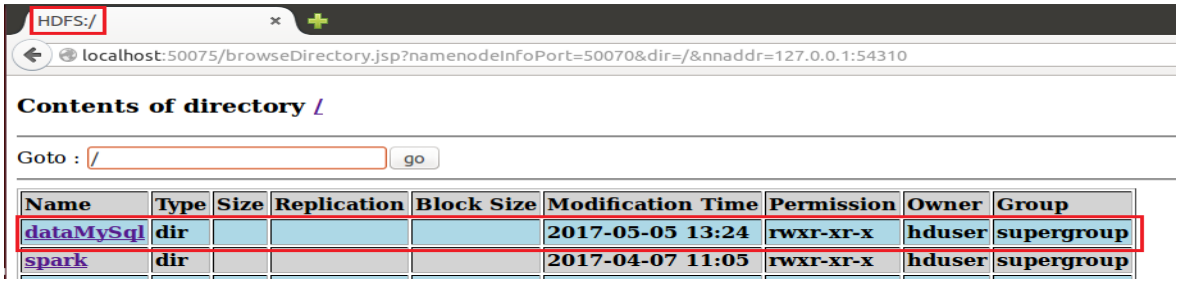


Figura 90: Carpeta de datos de MySQL a HDFS
Fuente: Elaboración propia.

Los datos de la tabla “data_consolidada_2v” de MySQL se encuentran en el archivo “part-m-00000” de la carpeta “dataMySql” y cada campo de la tabla se encuentra separado por pipes.

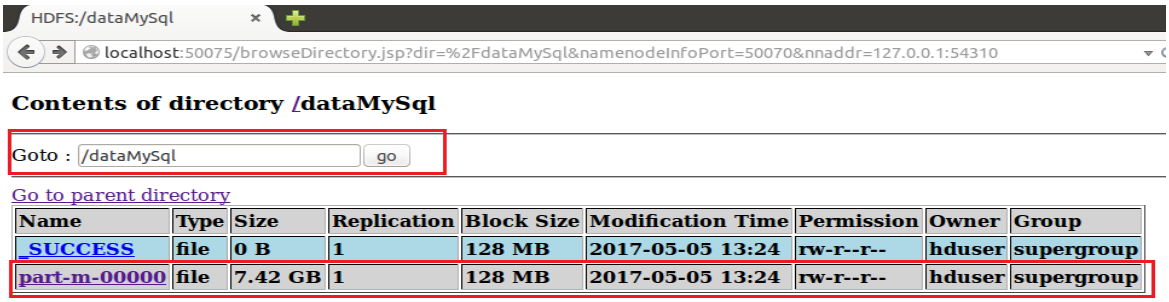


Figura 91: Archivo de datos de MySQL a HDFS
Fuente: Elaboración propia.

3.3.4. Funcionalidad y uso de Pig

3.3.4.1. Comandos básicos y sintaxis para Pig.

Apache Pig utiliza su propio lenguaje de programación llamado Pig Latín para la generación de comandos y sintaxis. Basándonos en lo que menciona (White, 2015) las instrucciones y comandos de Pig Latín operan en relaciones a los que se los denomina Operadores Relacionales. Cada operador tiene una funcionalidad similar a las instrucciones del lenguaje SQL, las cuales se mencionan a continuación:

Operadores Relacionales de Carga y Almacenamiento

Para tomar la información de un fichero o cualquier otro tipo de entrada de datos, se define qué datos se van a cargar, a esto Pig lo llama esquema. Pig facilita el proceso de tratamiento de datos estructurados y no estructurados.

LOAD

Carga un conjunto de datos desde el sistema de archivos de (HDFS) u otro almacenamiento, y los asocia a una relación.

```
<relacion> = LOAD <entrada> [ AS (<esquema>) ] [USING <funcion>]
```

Figura 92: Load en Pig
Fuente: Elaboración propia.

La opción <using> define qué función se usará para obtener los datos, si esta cláusula no es incluida, por defecto se utiliza PigStorage ('<Tabulador>'), el mismo que recoge datos PIG separados por tabuladores.

STORE

Recopila el resultado del código procesado en Pig en un lugar de almacenamiento especificado, pudiendo ser HDFS u otro almacenamiento local.

```
STORE <relacion> INTO <salida> [USING <funcion>]
```

Figura 93: Store en Pig
Fuente: Elaboración propia.

DUMP

Se utiliza para visualizar los datos y el esquema de los datos en sí generados.

```
DUMP <relacion>
```

Figura 94: Dump en Pig
Fuente: Elaboración propia.

Operadores Relacionales de Filtrado

FILTER

Permite seleccionar que registros se filtraran a partir de la evaluación de un predicado.

```
<relación> = FILTER <relación> BY <predicado>
```

Figura 95: Filter en Pig
Fuente: Elaboración propia.

DISTINCT

Elimina registros duplicados de una relación.

```
<relacion> = DISTINCT <relacion> BY [PARALLEL n]
```

Figura 96: Distinct en Pig
Fuente: Elaboración propia.

FOREACH - GENERATE

Agrega, remueve o modifica campos de una relación. Toma un conjunto de expresiones y las aplica a cada registro.

```
<relacion> = FOREACH <relacion> GENERATE <expresion> [AS <esquema>] [<expresion> [AS <esquema>]….]
```

Figura 97: Foreach - Generate en Pig
Fuente: Elaboración propia.

Operadores Relacionales de Ordenamiento

ORDER BY

Ordena los datos de una relación a partir de un campo.

```
<relacion> = ORDER <relacion> BY { * [ASC|DESC] | <campo> [ASC|DESC] [, <campo> [ASC|DESC] …] } [PARALLEL n]
```

Figura 98: Order by en Pig
Fuente: Elaboración propia.

Operadores de Diagnóstico

DESCRIBE

Retorna el esquema de una relación. Con el comando DESCRIBE se observa cómo se van cambiando los datos por cada sentencia ejecutada.

```
DESCRIBE <relacion>
```

Figura 99: Describe en Pig
Fuente: Elaboración propia.

3.3.4.2. Aplicación de los comandos y sintaxis de Pig en caso de estudio 2.

Para el caso de estudio 2, los datos que fueron descargados desde MySQL y almacenados en el HDFS de Hadoop, son procesados y analizados por Pig.

Verificación de la instalación de Hadoop

Antes de utilizar Pig, es necesario que se encuentre instalado y levantado el servicio de Hadoop en la máquina virtual. Esto se explica en el ANEXO 3 en el apartado llamado Verificación de la instalación de Hadoop.

Inicio de Pig

Posterior al inicio de Hadoop, se inicia Pig en la máquina virtual ejecutando el siguiente comando en la terminal del sistema:

```
$ pig
```

Figura 100: Inicio de Pig
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

```
hduser@diego-VirtualBox:/home/diego$ pig
17/05/05 16:41:03 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
17/05/05 16:41:03 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
17/05/05 16:41:03 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2017-05-05 16:41:04,037 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2017-05-05 16:41:04,058 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/diego/pig_1494020464036.log
2017-05-05 16:41:04,225 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/hduser/.pigbootup not found
2017-05-05 16:41:12,700 [main] WARN  org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
2017-05-05 16:41:12,772 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-05-05 16:41:12,839 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-05-05 16:41:12,839 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:54310
2017-05-05 16:41:19,000 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:54311
2017-05-05 16:41:19,276 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-8a731ab3-57b9-4136-8393-e36850c41de3
2017-05-05 16:41:19,290 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt>
```

Figura 101: Resultado de inicio de Pig
Fuente: Elaboración propia.

Cargar datos de HDFS a Pig

Para cargar los datos de HDFS a Pig se lo hace migrando la información hacia una relación llamada “extract_details” con los campos id, retweeted, source, created_at, text, retweet_count, filter_level, in_reply_to_screen_name (campos de la tabla de MySQL) ejecutando el siguiente comando en Pig de la terminal del sistema:

```
grunt> extract_details = LOAD '/dataMySql/part-m-00000' USING PigStorage('|') as (id:chararray,
retweeted:chararray, source:chararray, created_at:chararray, text:chararray,
retweet_count:chararray, filter_level:chararray, in_reply_to_screen_name:chararray);
```

Figura 102: Creación de relación en Pig
Fuente: Elaboración propia.

Manipulación de Big Data en Pig

Al tener la información en una relación de Pig que contiene datos de la tabla de MySQL; se ejecuta las consultas utilizando Pig Latín, que nos permitirán darnos información de tendencias de la segunda vuelta electoral 2017 de acuerdo a las sentencias de consulta que se ejecuten.

En las consultas que se va a realizar se encuentra un análisis de sentimientos y es necesario cargar en el HDFS un archivo que tendrá cada una de las palabras que mostrarían un sentimiento positivo, negativo o neutral en cada uno de los registros. La valoración se la dará en un diccionario de datos que tendrá la palabra y el sentimiento valorado en un número entero donde: 1 es positivo, -1 es negativo y 0 es neutral.

El archivo con las palabras y su valoración tiene como nombre "sentimientos.txt" y se muestra en la siguiente figura:

Figura 103: Archivo para análisis de sentimientos
Fuente: Elaboración propia.

En la máquina virtual el archivo de nombre "sentimiento.txt" está almacenado en la carpeta "Downloads" y se lo pasa al HDFS de Hadoop utilizando el siguiente comando:

```
$ hadoop fs -put /home/diego/Downloads/sentimientos.txt /
```

Figura 104: Sentencia para pasar el archivo "sentimiento.txt" a HDFS
Fuente: Elaboración propia.

En el HDFS de Hadoop se observa el archivo "sentimientos.txt" que se ha copiado:

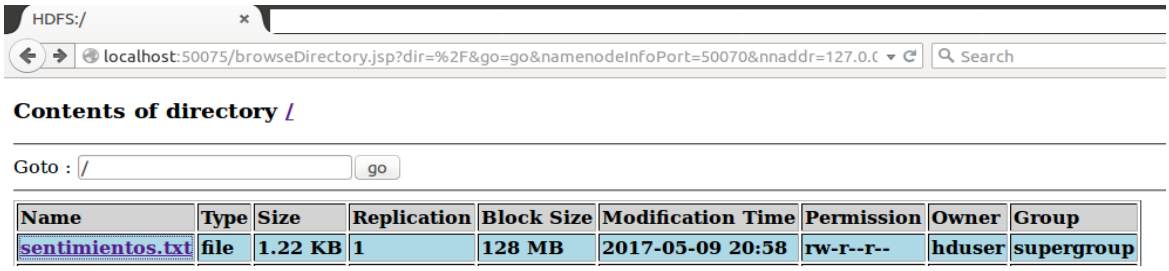


Figura 105: Archivo "sentimientos.txt" en HDFS
Fuente: Elaboración propia.

Se ha realizado la búsqueda de información en 5 consultas, las cuales han arrojado los siguientes datos:

- **Total de registros en 2da vuelta**

```
grunt> conta_Total = FOREACH (GROUP extract_details ALL) GENERATE
COUNT(extract_details);
grunt> dump conta_Total;
```

Figura 106: Sentencia para total de registros en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

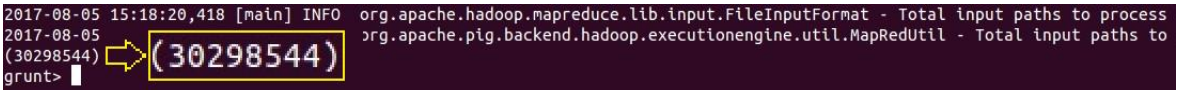


Figura 107: Resultado del total de registros en 2da vuelta
Fuente: Elaboración propia.

- **Cantidad de veces que se mencionaron a los candidatos en 2da vuelta**

```
grunt> tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word;
```

Figura 108: Sentencia para cargar en relación los registros separados por palabras
Fuente: Elaboración propia.

```
grunt> datos_lenin = FILTER tokens BY word == '@Lenin';
grunt> conta_lenin = FOREACH (GROUP datos_lenin ALL) GENERATE COUNT(datos_lenin);
grunt> dump conta_lenin;
```

Figura 109: Sentencia para total de registros relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

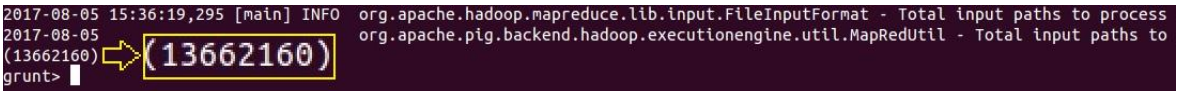


Figura 110: Resultado del total de registros relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

```
grunt> datos_lasso = FILTER tokens BY word == '@LassoGuillermo';
grunt> conta_lasso = FOREACH (GROUP datos_lasso ALL) GENERATE COUNT(datos_lasso);
grunt> dump conta_lasso;
```

Figura 111: Sentencia para total de registros relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

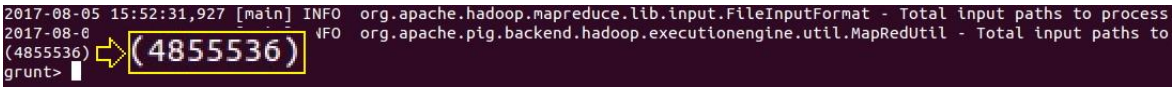


Figura 112: Resultado del total de registros relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

- **Cantidad de veces mencionando Rafael Correa a los candidatos en 2da vuelta**

```
grunt> tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word;
```

Figura 113: Sentencia para cargar en relación los registros separados por palabras
Fuente: Elaboración propia.

```
grunt> datos_rafael = FILTER tokens BY word == '@MashiRafael:';
grunt> conta_rafael = FOREACH (GROUP datos_rafael ALL) GENERATE COUNT(datos_rafael);
grunt> dump conta_rafael;
```

Figura 114: Sentencia para total de registros publicados por @MashiRafael en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

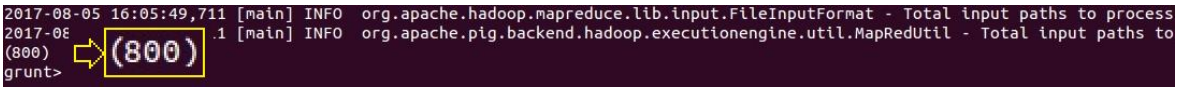


Figura 115: Resultado del total de registros publicados por @MashiRafael en 2da vuelta
Fuente: Elaboración propia.

- **Análisis de sentimientos hacia @Lenin en 2da vuelta**

```
grunt> tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word;
```

Figura 116: Sentencia para cargar en relación los registros separados por palabras
Fuente: Elaboración propia.

```
grunt> dictionary = load '/sentimientos.txt' using PigStorage('\t') AS(word:chararray,rating:int);
```

Figura 117: Sentencia para cargar en relación el diccionario de sentimientos
Fuente: Elaboración propia.

```
grunt> datos_lenin = FILTER tokens BY word == '@Lenin';
grunt> tweets_lenin = foreach datos_lenin generate id,text;
grunt> tokens_lenin = foreach tweets_lenin generate id,text, FLATTEN(TOKENIZE(text)) As word;
grunt> word_rating_lenin = join tokens_lenin by word left outer, dictionary by word using
'replicated';
grunt> rating_lenin = foreach word_rating_lenin generate tokens_lenin::id as id,tokens_lenin::text
as text, dictionary::rating as rate;
grunt> word_group_lenin = group rating_lenin by (id,text);
grunt> avg_rate_lenin = foreach word_group_lenin generate group, AVG(rating_lenin.rate) as
tweet_rating;
```

Figura 118: Sentencia para consultar sentimientos de registros @Lenin en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_positivos_lenin = filter avg_rate_lenin by tweet_rating>0.5;
grunt> conta_tweets_positivos_lenin = FOREACH (GROUP tweets_positivos_lenin ALL)
GENERATE COUNT(tweets_positivos_lenin);
grunt> dump conta_tweets_positivos_lenin;
```

Figura 119: Sentencia para sentimientos positivos relacionados a @Lenin en Pig
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

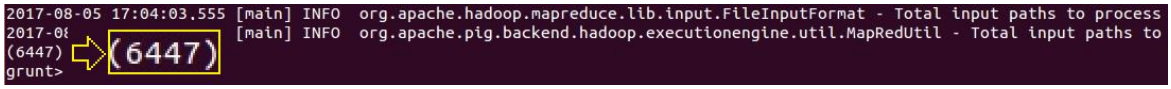


Figura 120: Resultado del total de sentimientos positivos relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_negativos_lenin = filter avg_rate_lenin by tweet_rating<-0.5;
grunt> conta_tweets_negativos_lenin = FOREACH (GROUP tweets_negativos_lenin ALL)
GENERATE COUNT(tweets_negativos_lenin);
grunt> dump conta_tweets_negativos_lenin;
```

Figura 121: Sentencia para sentimientos negativos relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

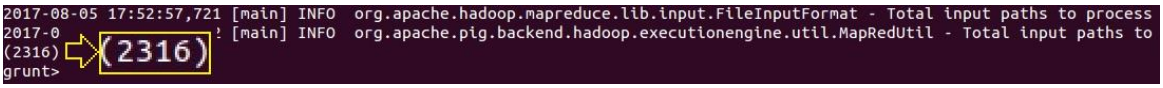


Figura 122: Resultado del total de sentimientos negativos relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_neutral_lenin = filter avg_rate_lenin by ((tweet_rating>-0.5) AND
(tweet_rating<0.5));
grunt> conta_tweets_neutral_lenin = FOREACH (GROUP tweets_neutral_lenin ALL) GENERATE
COUNT(tweets_neutral_lenin);
grunt> dump conta_tweets_neutral_lenin;
```

Figura 123: Sentencia para sentimientos neutrales relacionados a @Lenin en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

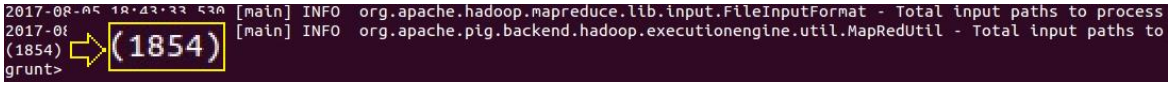


Figura 124: Resultado del total de sentimientos neutrales relacionados a @Lenin en Pig
Fuente: Elaboración propia.

- **Análisis de sentimientos hacia @LassoGuillermo en 2da vuelta**

```
grunt> tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word;
```

Figura 125: Sentencia para cargar en relación los registros separados por palabras
Fuente: Elaboración propia.

```
grunt> dictionary = load '/sentimientos.txt' using PigStorage('\t') AS(word:chararray,rating:int);
```

Figura 126: Sentencia para cargar en relación el diccionario de sentimientos
Fuente: Elaboración propia.

```
grunt> datos_lasso = FILTER tokens BY word == '@LassoGuillermo';
grunt> tweets_lasso = foreach datos_lasso generate id,text;
grunt> tokens_lasso = foreach tweets_lasso generate id,text, FLATTEN(TOKENIZE(text)) As
word;
grunt> word_rating_lasso = join tokens_lasso by word left outer, dictionary by word using
'replicated';
grunt> rating_lasso = foreach word_rating_lasso generate tokens_lasso::id as
id,tokens_lasso::text as text, dictionary::rating as rate;
grunt> word_group_lasso = group rating_lasso by (id,text);
grunt> avg_rate_lasso = foreach word_group_lasso generate group, AVG(rating_lasso.rate) as
tweet_rating;
```

Figura 127: Sentencia para consultar sentimientos de registros @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_positivos_lasso = filter avg_rate_lasso by tweet_rating>0.5;
grunt> conta_tweets_positivos_lasso = FOREACH (GROUP tweets_positivos_lasso ALL)
GENERATE COUNT(tweets_positivos_lasso);
grunt> dump conta_tweets_positivos_lasso;
```

Figura 128: Sentencia para sentimientos positivos relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

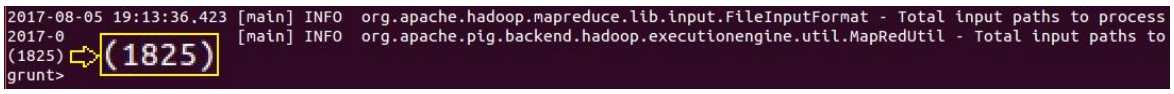


Figura 129: Resultado total de sentimientos positivos relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_negativos_lasso = filter avg_rate_lasso by tweet_rating<-0.5;
grunt> conta_tweets_negativos_lasso = FOREACH (GROUP tweets_negativos_lasso ALL)
GENERATE COUNT(tweets_negativos_lasso);
grunt> dump conta_tweets_negativos_lasso;
```

Figura 130: Sentencia para sentimientos negativos relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

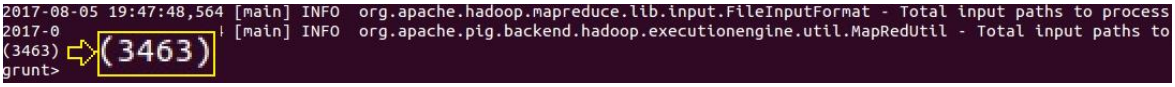


Figura 131: Resultado total de sentimientos negativos relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

```
grunt> tweets_neutral_lasso = filter avg_rate_lasso by ((tweet_rating>-0.5) AND
(tweet_rating<0.5));
grunt> conta_tweets_neutral_lasso = FOREACH (GROUP tweets_neutral_lasso ALL)
GENERATE COUNT(tweets_neutral_lasso);
grunt> dump conta_tweets_neutral_lasso;
```

Figura 132: Sentencia para sentimientos neutrales relacionados a @LassoGuillermo en Pig
Fuente: Elaboración propia.

En la máquina virtual, el resultado de la consulta es el siguiente:

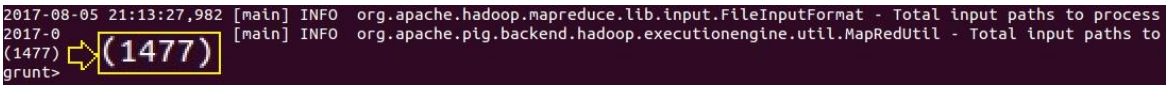


Figura 133: Resultado total de sentimientos neutrales relacionados a @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

CAPÍTULO 4: ANÁLISIS DE LAS 4 HERRAMIENTAS SELECCIONADAS DE HADOOP

4.1. Análisis e interpretación de resultados y datos

4.1.1. Resultados del caso de estudio 1

Al obtener los resultados después de usar las herramientas Flume y Hive en el caso de estudio 1, es necesario analizar e interpretar los datos arrojados. Esto se lo hace a continuación.

4.1.1.1. Total de registros en 1ra vuelta

La primera vuelta electoral dio como inicio el 3 de enero del 2017 y finalizó el día de las elecciones el 19 de febrero del 2017. Durante todo ese tiempo la red social Twitter estuvo activa constantemente, de la cual se descargó con la herramienta Flume un total de **131512320** twitters que fueron generados en diferentes horas del día en todo el territorio ecuatoriano. De los twitters descargados, un total de **28055040** tweets hacen relación a los candidatos de la primera vuelta electoral.

Esta relación se lo muestra en la siguiente tabla:

Tabla 7: Total de datos descargados de Twitter 1ra vuelta

Detalle	Cantidad
Tweets otros temas	103457280
Tweets relacionado a candidatos	28055040
TOTAL	131512320

Fuente: Elaboración propia.

En la figura 133 se observa la relación en porcentaje de la tabla 7:

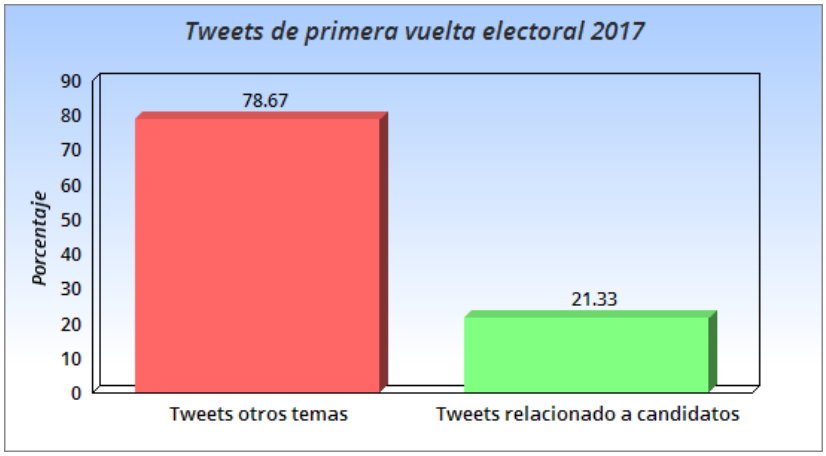


Figura 134: Porcentaje tweets de 1ra vuelta
Fuente: Elaboración propia.

Interpretación: Durante el periodo de la primera vuelta electoral, en la red social Twitter las publicaciones que realizaron varias personas en Ecuador abarcaron un total de 131512320

tweets el cual representa el 100% de información obtenida. De ese porcentaje el 78.67% (103457280) son tweets relacionados a varios temas diferentes a política y un 21.33% (28055040) de los tweets hablaron de los candidatos presidenciales que fueron parte de la primera vuelta electoral 2017. Muchas de las personas han utilizado la red social para dar a conocer sus comentarios por algún candidato, y esto es un buen punto de partida para explotar la información utilizando Hadoop.

4.1.1.2. Número de veces que se mencionaron a los candidatos en 1ra vuelta.

De los 28055040 tweets en los que se habla de los candidatos presidenciales, se realiza una separación de la cantidad de veces que son mencionados cada uno de los candidatos presidenciales de la primera vuelta. El resultado de esta separación se lo muestra en la siguiente tabla:

Tabla 8: Cantidad de tweets que se mencionaron a los candidatos en 1ra vuelta

Candidato	Cantidad de veces
@Lenin	10813568
@LassoGuillermo	10083328
@CynthiaViteri6	3928448
@PacoMoncayo	1716480
@daloes10	1205376
@IvanEspinelM	167040
@pesanteztwof	77824
@ZuquilandaDuque	62976
TOTAL:	28055040

Fuente: Elaboración propia.

En la figura 134 se observa la relación en porcentaje de la tabla 8:

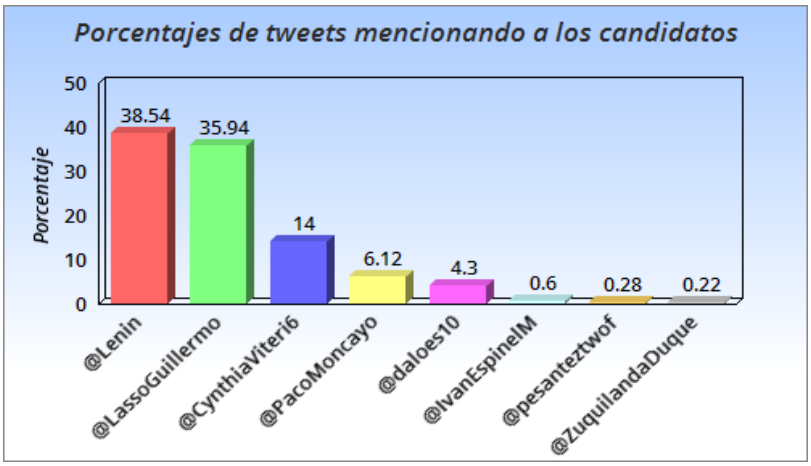


Figura 135: Porcentajes de tweets que se mencionaron a los candidatos en 1ra vuelta
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados, se establece que los 2 principales candidatos que son los más comentados en Twitter son Lenín Moreno y Guillermo Lasso. Esto sirve como base para proyectar que estos dos candidatos han de continuar a la segunda vuelta electoral.

4.1.1.3. Cantidad de tweets que publicó cada candidato en 1ra vuelta.

De los 28055040 tweets en los que se habla de los candidatos presidenciales, se realiza una separación de la cantidad de veces que estos candidatos realizaron publicaciones en la red social Twitter durante la primera vuelta electoral. El resultado de esta separación dio como resultado 14190 tweets y se los observa en la siguiente tabla:

Tabla 9: Cantidad de tweets de los candidatos presidenciales en 1ra vuelta

Candidato	Cantidad de veces
@Lenin	6095
@LassoGuillermo	1106
@CynthiaViteri6	1095
@PacoMoncayo	2457
@daloes10	1550
@IvanEspinelM	1126
@pesanteztwof	576
@ZuquilandaDuque	185
TOTAL:	14190

Fuente: Elaboración propia.

En la figura 135 se observa la relación en porcentaje de la tabla 9:

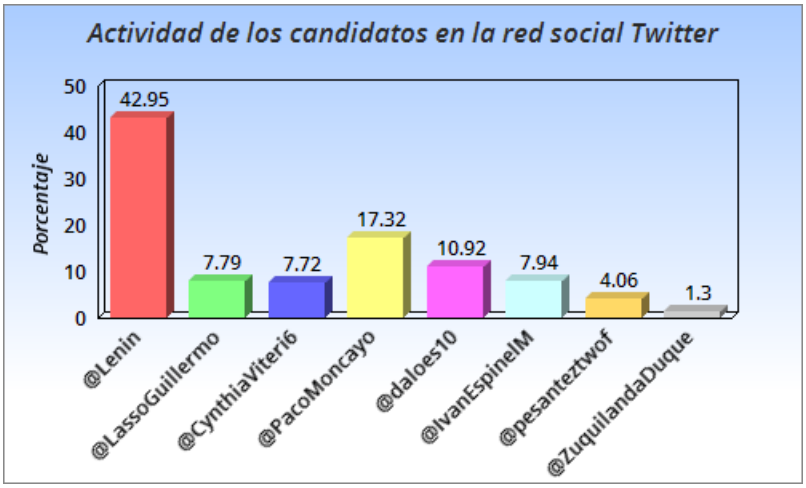


Figura 136: Porcentajes de tweets publicados por los candidatos presidenciales
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados, se establece que Lenín Moreno es el candidato más activo en la red social Twitter, mientras que Patricio Zuquilanda es el candidato menos activo durante la primera vuelta electoral 2017.

4.1.1.4. Cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta.

De los 28055040 tweets en los que se habla de los candidatos presidenciales, se efectúa una separación de la cantidad de veces que el presidente Rafael Correa realiza publicaciones en la red social Twitter durante la primera vuelta electoral 2017 mencionando a los candidatos. El resultado de esta separación da como resultado 520 tweets y se lo observa en la siguiente tabla:

Tabla 10: Cantidad de tweets de Rafael Correa por candidatos en 1ra vuelta

Candidato	Cantidad de veces
@Lenin	101
@LassoGuillermo	120
@CynthiaViteri6	88
@PacoMoncayo	51
@daloes10	108
@IvanEspinelM	28
@pesanteztwof	12
@ZuquilandaDuque	12
TOTAL:	520

Fuente: Elaboración propia.

En la figura 136 se observa la relación en porcentaje de la tabla 10:

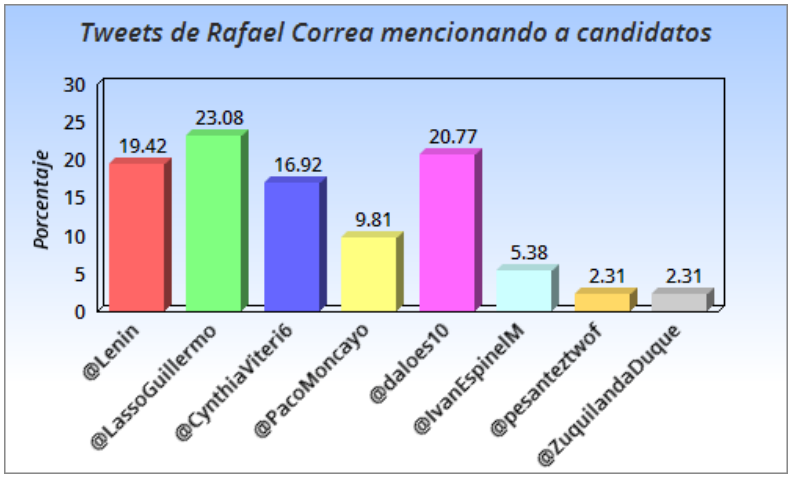


Figura 137: Porcentajes de tweets de Rafael Correa mencionando a candidatos
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados, se establece que Rafael Correa realiza la mayor parte de sus comentarios durante la primera vuelta electoral 2017 mencionando al candidato Guillermo Lasso y en menor cantidad comentando acerca de Washington Pesantez y Patricio Zuquilanda.

4.1.1.5. Tweets que han sido más compartidos o retweeteados en 1ra vuelta.

De los 131512320 tweets descargados durante la primera vuelta electoral, se ejecuta una separación de cuáles son los tweets que más se compartieron (mayores a 200000 veces). El resultado de esta separación muestra como resultado 8 tweets más retweeteados y que al sumarlos dieron un total de 3268234 veces compartidos; esto se lo observa en la siguiente tabla 11:

Tabla 11: Cantidad de tweets que han sido más compartidos o retweeteados

Número	Texto	No. Compartidos
Texto 1	RT @AlejoCassola: Gracias @CynthiaViteri6 y @PacoMoncayo, sin ustedes la victoria de @Lenin no estaría tan cerca.	319616
Texto 2	RT @Kary_Arteaga: Este 19 de febrero: Cuento con tu voto@Lenin	259500
Texto 3	RT @MrVertigo7: @LassoGuillermo dic q está en 2da vuelta #EleccionesEnEcuador Pero hay que esperar datos oficiales de @cnegobec	266624
Texto 4	RT @PLJV7: #NebotTraidor #Ni1VotoParaLa6 #VigiliaElectoral Centro de Convenciones @CynthiaViteri6 @jaimenebotsaadi PSC=AP..	315008
Texto 5	RT @VamosLenin: .@Lenin: "Queridos jóvenes, la política es el arte de servir. Que pena que ciertos políticos la hayan hecho fea".	471684
Texto 6	RT @VamosLenin: Las necesidades de los jóvenes serán escuchadas y atendidas por el próximo presidente del #19F: @Lenin Moreno.	562881
Texto 7	RT @carlitoswayec: Pase o no @LassoGuillermo a 2a vuelta, la historia pondrá en su sitio a todos y a cada uno d los chimbadores pero sobre to…	666368
Texto 8	RT @veroecua: 19 F.Vota 35/ #AlainAsambleista ¡Por los emprendedores de nuestra Patria! @AlainVelez @35PAIS #Distrito3…	406553
	TOTAL:	3268234

Fuente: Elaboración propia.

En la figura 137 se observa la relación en porcentaje de la tabla 11:

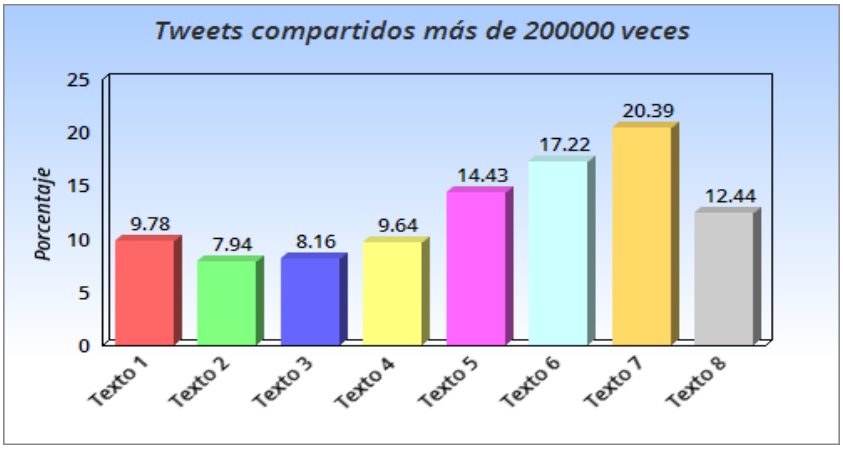


Figura 138: Porcentajes de tweets más compartidos
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados mostrados en la figura 137, se establece que el texto 7 es el más compartido y el texto 2 es el menos compartido, ambos en un rango mayor a 200000. Estos resultados sirven de base para futuros análisis dependiendo del alcance que se le quiera dar a la información proporcionada. Ejemplo: Análisis de sentimientos, estadísticas, etc.

4.1.2. Resultados del caso de estudio 2

Al obtener los resultados después de usar las herramientas Sqoop y Pig en el caso de estudio 2, es necesario analizar e interpretar los datos arrojados. Esto se lo hace a continuación.

4.1.2.1. Total de registros en 2da vuelta

La segunda vuelta electoral dio inicio el 10 de marzo del 2017 y finalizó el día de las elecciones 02 de abril del 2017. Durante todo ese tiempo la red social Twitter estuvo activa constantemente, de la cual se descargó tweets que fueron almacenados en tablas relacionales de MySQL mediante un procesamiento de datos. Para facilitar su uso se creó y se almacenó en una sola tabla la información de **30298544** registros los cuales hacen relación a los datos de Twitter que se encontraban en las tablas relacionales antes mencionadas. Posteriormente, usando Sqoop los registros de la tabla de MySQL fueron migrados a Pig. De estos registros, un total de **18517696** hacen relación a los candidatos de la segunda vuelta electoral. Esta relación se lo muestra en la siguiente tabla:

Tabla 12: Total de registros descargados de Twitter 2da vuelta

Detalle	Cantidad
Total registros otros temas	11780848
Total registros relacionado a candidatos	18517696
TOTAL:	30298544

Fuente: Elaboración propia.

En la figura 138 se observa la relación en porcentaje de la tabla 12:

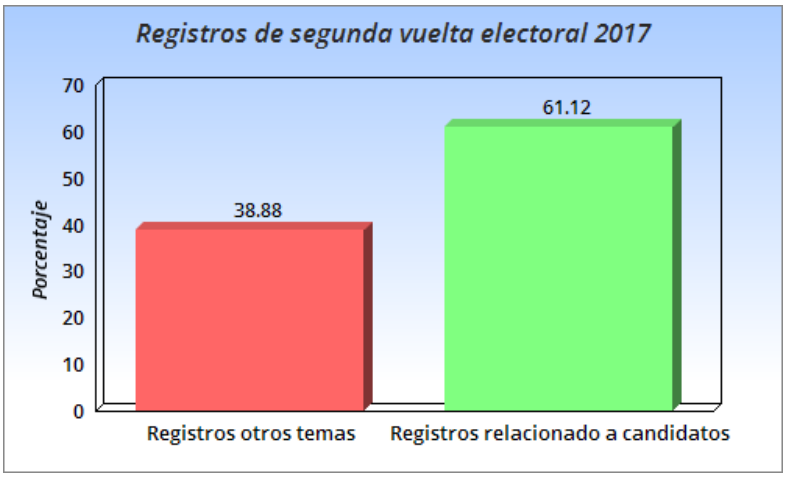


Figura 139: Porcentaje registros descargados de Twitter 2da vuelta
Fuente: Elaboración propia.

Interpretación: Durante el periodo de la segunda vuelta electoral, en la red social Twitter las publicaciones que realizaron varias personas en Ecuador abarcaron un total de 30298544 tweets el cual representa el 100% de información obtenida y almacenada en MySQL. De ese porcentaje el 39% (11780848) son tweets relacionados a varios temas diferentes a la política y un 61% (18517696) de los tweets hablaron de los candidatos presidenciales que son parte de la segunda vuelta electoral. Los comentarios y publicaciones en la segunda vuelta electoral en la red social Twitter son más activos en comparación a la primera vuelta electoral 2017.

4.1.2.2. Cantidad de veces que se mencionaron a los candidatos en 2da vuelta

De los 18517696 registros en los que se habla de los candidatos presidenciales, se realiza una separación de la cantidad de veces que fueron mencionados cada uno de los candidatos en la segunda vuelta electoral 2017. El resultado de esta separación se lo muestra en la siguiente tabla:

Tabla 13: Cantidad de registros que se mencionaron a los candidatos en 2da vuelta

Candidato	Cantidad de veces
@Lenin	13662160
@LassoGuillermo	4855536
TOTAL:	18517696

Fuente: Elaboración propia.

En la figura 139 se observa la relación en porcentaje de la tabla 13:

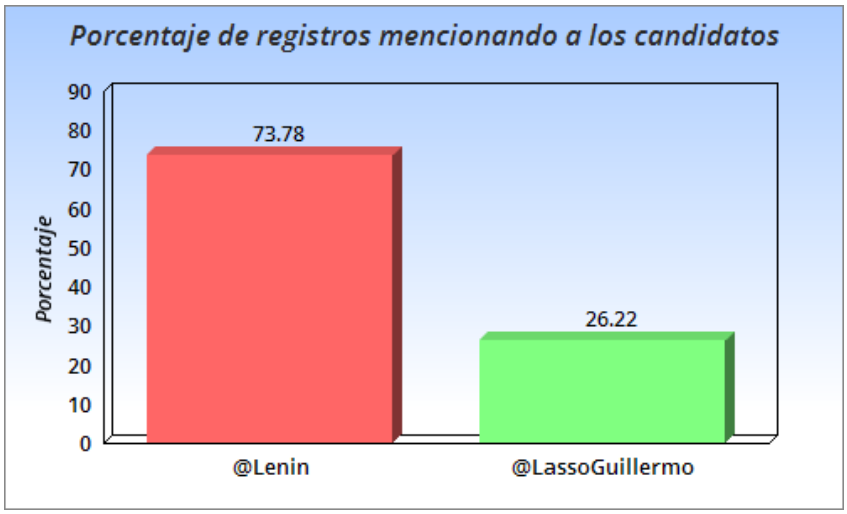


Figura 140: Porcentaje de registros mencionando a los candidatos en 2da vuelta
Fuente: Elaboración propia.

Interpretación: Basándonos en el porcentaje de los resultados, se establece que el principal candidato de quién más se comentó es Lenín Moreno. Esto sirve como base para proyectar que tiene más probabilidad de ser el presidente del Ecuador.

4.1.2.3. Cantidad de veces mencionando Rafael Correa a los candidatos en 2da vuelta

De los 30298544 registros almacenados en MySQL y procesados en Pig en los que se habla de los 2 candidatos presidenciales y varios temas, se realiza una separación de la cantidad de veces que el presidente Rafael Correa realizó publicaciones en la red social Twitter durante la segunda vuelta electoral. El resultado de esta separación dio como resultado 800 registros y se lo observa en la siguiente tabla:

Tabla 14: Cantidad de registros de Rafael Correa en 2da vuelta

Detalle	Cantidad
Registros de @MashiRafael	800
Otros usuarios	30297744
TOTAL:	30298544

Fuente: Elaboración propia.

En la figura 140 se observa la relación en porcentaje de la tabla 14:

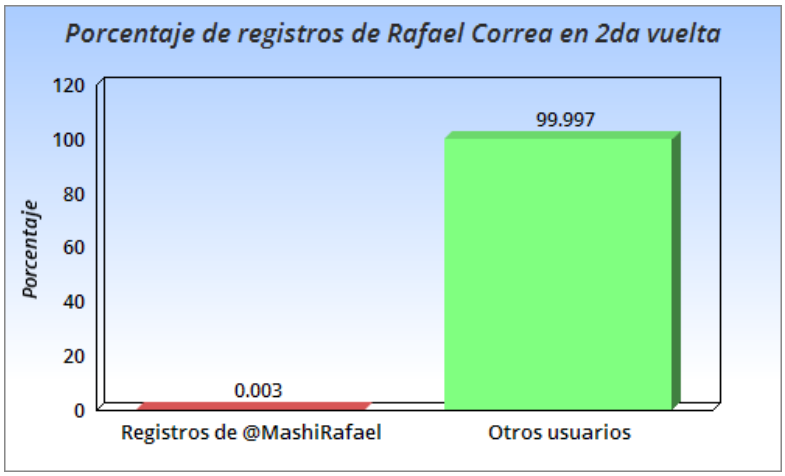


Figura 141: Porcentajes de registros de Rafael Correa en 2da vuelta
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados, se establece que Rafael Correa realiza más comentarios durante la segunda vuelta electoral que en la primera vuelta electoral 2017.

4.1.2.4. Análisis de sentimientos hacia @Lenin en 2da vuelta

De los 18517696 registros de la segunda vuelta electoral en los que se habla de los candidatos presidenciales en MySQL procesados por Pig, se realiza una separación de los registros que demuestren sentimiento emocional hacia el candidato presidencial Lenín Moreno. En total se contabilizaron 10617 registros en los que constan sentimientos positivos, negativos y neutrales. El resultado de esta separación se lo muestra en la siguiente tabla:

Tabla 15: Cantidad de sentimientos por @Lenin en 2da vuelta

Sentimiento	Cantidad
Positivo	6447
Negativo	2316
Neutral	1854
TOTAL:	10617

Fuente: Elaboración propia.

En la figura 141 se observa la relación en porcentaje de la tabla 15:

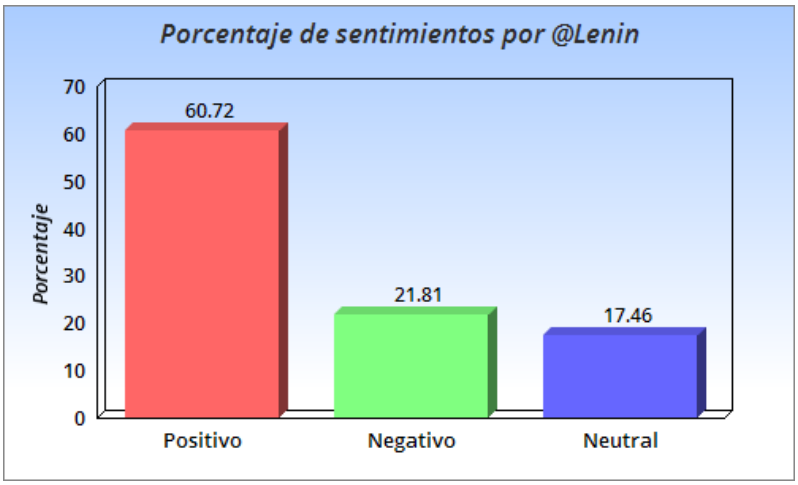


Figura 142: Porcentaje de sentimientos por @Lenin en 2da vuelta
Fuente: Elaboración propia.

Interpretación: Basándonos en el porcentaje de los resultados, se establece que la mayor parte de comentarios realizados son positivos a favor del candidato Lenín Moreno. Si se mantiene esa tendencia se proyecta que tiene una alta probabilidad de ser el presidente del Ecuador.

4.1.2.5. Análisis de sentimientos hacia @LassoGuillermo en 2da vuelta

De los 4855536 registros de la segunda vuelta electoral en los que se habla de los candidatos presidenciales en MySQL procesados por Pig, se realiza una separación de los registros que demuestren sentimiento emocional hacia el candidato presidencial Guillermo Lasso. En total se contabilizaron 6765 registros en los que constan sentimientos positivos, negativos y neutrales. El resultado de esta separación se lo muestra en la siguiente tabla:

Tabla 16: Cantidad de sentimientos por @LassoGuillermo en 2da vuelta

Sentimiento	Cantidad
Positivo	1825
Negativo	3463
Neutral	1477
TOTAL:	6765

Fuente: Elaboración propia.

En la figura 142 se observa la relación en porcentaje de la tabla 16:

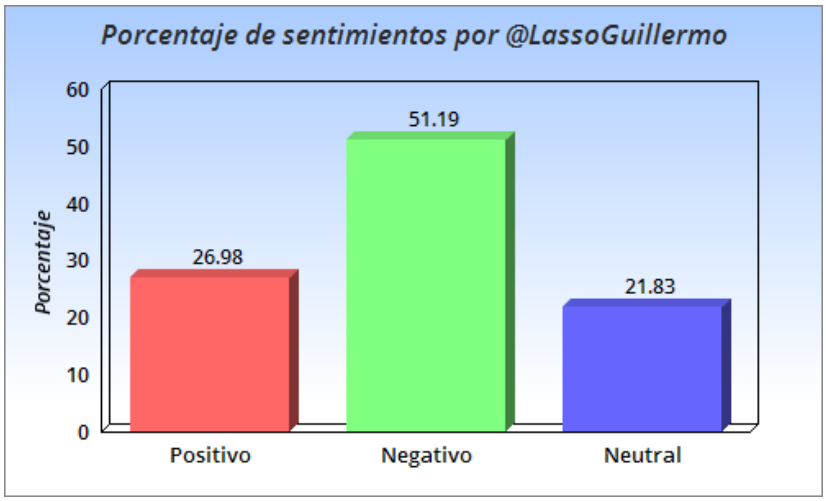


Figura 143: Porcentaje de sentimientos por @LassoGuillermo
Fuente: Elaboración propia.

Interpretación: Basándonos en los porcentajes de los resultados, se establece que la mayor parte de comentarios realizados son negativos para el candidato Guillermo Lasso. Si se mantiene esa tendencia se proyecta que tiene menos probabilidad de ser el presidente del Ecuador.

4.2. Análisis comparativo de las herramientas seleccionadas

4.2.1. Análisis comparativo para determinar ventajas y desventajas

Al seleccionar y manejar las herramientas de Hadoop que forman parte del presente trabajo, es necesario realizar un análisis comparativo en el que se analice información general, funcionalidad y usabilidad de las 4 herramientas; esto con el fin de describir características que permitan determinar ventajas y desventajas que tengan entre sí.

4.2.1.1. Análisis comparativo general

En este análisis comparativo general se busca establecer la información técnica y básica de las principales características que son necesarias al analizar a las 4 herramientas de Hadoop. En la siguiente tabla se describen las características generales que evalúan ventajas y desventajas por cada una de las 4 herramientas de Hadoop:

Tabla 17: Análisis comparativo general

CARACTERÍSTICAS	HERRAMIENTAS			
	Flume	Hive	Sqoop	Pig

Instalación y configuración	- Se necesita contar con la instalación previa de Hadoop. - Debe estar iniciado Hadoop en la máquina local. - Su instalación en la máquina local es fácil. Existen tutoriales fáciles de aplicar. - Su configuración tiene un grado medio de dificultad, ya que existen fuentes de datos para recolectar la información.	- Se necesita contar con la instalación previa de Hadoop. - Debe estar iniciado Hadoop en la máquina local. - Su instalación en la máquina local es difícil. Hive almacena sus metadatos en una base de datos Derby y su implementación tiene su respectiva dificultad. - Su configuración tiene un grado medio de dificultad, ya que cada propiedad de configuración que no esté correctamente configurada lanzará una excepción.	- Se necesita contar con la instalación previa de Hadoop. - Debe estar iniciado Hadoop en la máquina local. - Su instalación en la máquina local es fácil. Existen tutoriales fáciles de aplicar. - Es de fácil configuración. Son pocas propiedades que se deben configurar en Hadoop.	- Se necesita contar con la instalación previa de Hadoop. - No es necesario iniciar Hadoop, se ejecuta también en modo autónomo. - Su instalación en la máquina local es fácil. Existen tutoriales fáciles de aplicar. - Es de fácil configuración. Son pocas propiedades que se deben aplicar en Hadoop.
Licencia	- Licencia de software libre escrita por Apache Software Foundation. Permite al usuario del software la libertad para usarlo, modificarlo y distribuir versiones modificadas. - La licencia Apache no obliga a que las modificaciones se deban distribuir como software libre.	- Licencia de software libre escrita por Apache Software Foundation. Permite al usuario del software la libertad para usarlo, modificarlo y distribuir versiones modificadas. - La licencia Apache no obliga a que las modificaciones se deban distribuir como software libre.	- Licencia de software libre escrita por Apache Software Foundation. Permite al usuario del software la libertad para usarlo, modificarlo y distribuir versiones modificadas. - La licencia Apache no obliga a que las modificaciones se deban distribuir como software libre.	- Licencia de software libre escrita por Apache Software Foundation. Permite al usuario del software la libertad para usarlo, modificarlo y distribuir versiones modificadas. - La licencia Apache no obliga a que las modificaciones se deban distribuir como software libre.
Costo de licencia	- No tiene costo	- No tiene costo	- No tiene costo	- No tiene costo

Enfocado al trabajo con Big Data	- Su funcionamiento está enfocado al trabajo con Big Data.	- Su funcionamiento está enfocado al trabajo con Big Data.	- Su funcionamiento está enfocado al trabajo con Big Data.	- Su funcionamiento está enfocado al trabajo con Big Data.
Integración con las herramientas del ecosistema de Hadoop	- Se integra correctamente con Hadoop y con otras herramientas de Hadoop. - Apache Flume requiere ser instalado no sólo en el clúster Hadoop, sino en cada una de las máquinas que almacenan o generan los logs, en las que se ejecuta un denominado agente de Apache Flume encargado de la recolección de nueva información.	- Se relaciona correctamente con Hadoop y otras herramientas de Hadoop. - Intercambia y procesar datos con otras herramientas de Hadoop.	- Se integra correctamente con Hadoop y otras herramientas de Hadoop. - Obtiene y almacenar diferentes datos en las herramientas de Hadoop.	- Se relaciona correctamente con Hadoop y otras herramientas de Hadoop. - Procesa datos de varias herramientas de Hadoop.
Ayuda al usuario	- La herramienta cuenta con ayuda al usuario para la ejecución de comandos que facilitan el procesamiento de la información.	- La herramienta cuenta con ayuda al usuario para la ejecución de comandos que facilitan el procesamiento de la información.	- La herramienta cuenta con ayuda al usuario para la ejecución de comandos que facilitan el procesamiento de la información.	- La herramienta cuenta con ayuda al usuario para la ejecución de comandos que facilitan el procesamiento de la información.
Soporte técnico	- Tiene soporte para el uso de la aplicación por parte de Apache Software Foundation. - Muchos de los tutoriales de usuarios brindan ayuda y dan soporte al manejo de la aplicación.	- Tiene soporte para el uso de la aplicación por parte de Apache Software Foundation.	- Tiene soporte para el uso de la aplicación por parte de Apache Software Foundation. - Muchos de los tutoriales de usuarios brindan ayuda y dan soporte al manejo de la aplicación.	- Tiene soporte para el uso de la aplicación por parte de Apache Software Foundation.
Manejo de Interfaces y gráficos	- No. El uso de la herramienta se lo hace a través de líneas de	- No. El uso de la herramienta se lo hace a través de líneas de	- No. El uso de la herramienta se lo hace a través de líneas de	- No. El uso de la herramienta se lo hace a través de líneas de

	comando.	comando.	comando.	comando.

Fuente: Elaboración propia.

4.2.1.2. Análisis comparativo de funcionalidad

En este análisis comparativo de funcionalidad se busca establecer la información relacionada al funcionamiento que realizan las 4 herramientas de Hadoop para el procesamiento de Big Data. En la tabla 18 se describen las características generales que evalúan ventajas y desventajas comparando su funcionalidad:

Tabla 18: Análisis comparativo de funcionalidad

CARACTERÍSTICAS	**HERRAMIENTAS**			
	Flume	**Hive**	**Sqoop**	**Pig**
Tipo de procesamiento de la información	- Data integration: suministra datos a Hadoop procedente de otras fuentes de datos o la extracción de la información del clúster de Hadoop a otros repositorios.	- Data Analysis: manejo de datos existentes en Hadoop utilizando algún tipo de lenguaje de consulta que generan procesos Map Reduce que se ejecutan en el clúster Hadoop.	- Data integration: suministra datos a Hadoop procedente de otras fuentes de datos o la extracción de la información del clúster de Hadoop a otros repositorios.	- Data Analysis: manejo de datos existentes en Hadoop utilizando algún tipo de lenguaje de consulta que generan procesos Map Reduce que se ejecutan en el clúster Hadoop.
Análisis de datos y metadatos	- No. Su uso se limita a la recolección de datos de diferentes fuentes y almacenarlos en el HDFS de Hadoop.	- Si. Mediante la creación de una base de datos similar a un modelo Entidad - Relación.	- No. Utilizado para transferir de forma eficiente datos entre Hadoop y bases de datos relacionales.	- No. Utiliza su propio lenguaje de procesamiento para el análisis de datos pero no brinda soporte de metadatos.
Lenguaje para manejo de datos	- No.	- Si. Permite el manejo y utilización de datos almacenas en Hadoop a través del lenguaje tipo SQL denominado HiveQL.	- No.	- Si. Utiliza su propio lenguaje de manejo de datos llamado Pig Latín - Pig Latín facilita la creación de programas MapReduce utilizados para el procesamiento de los datos almacenados en una infraestructura Hadoop.

Concurrencia de las herramientas en máquina local	- No. Maneja un solo archivo de configuración para la recolección de datos. Al ser único no permite concurrencia.	- No. La herramienta utiliza un solo servicio en Hadoop el cual no permite ejecutar varias terminales de Hive.	- No. La herramienta bloquea el uso de los comandos Sqoop mientras se este previamente procesando datos con Sqoop.	- Si. Permite manejar concurrencia para ejecutar diferentes programas MapReduce independientemente una de otra.
Manejo de grandes volúmenes de información	- Si. La descarga de datos lo hace masivamente desde diferentes fuentes de datos.	- Si. Utiliza procesos MapReduce para el procesamiento de grandes volúmenes de datos.	- Si. Mueve grandes volúmenes de datos desde base de datos relacionales.	- Si. Maneja grandes volúmenes de datos mediante su propio lenguaje de scripts llamado Pig Latín.
Tiempo de respuesta al procesamiento de datos	- El tiempo de respuesta para la descarga de datos es medio. Dependerá de la configuración para recolectar información.	- El tiempo de respuesta es alto. Los tiempos de respuesta son mucho más rápido que otros tipos de consultas sobre el mismo tipo de conjuntos de datos.	- El tiempo de respuesta es alto. Los datos son movidos rápidamente. Dependerá de cantidad de datos a procesar.	- El tiempo de respuesta es alto. Aunque se necesita de codificación de tareas para el procesamiento de datos.

Fuente: Elaboración propia.

4.2.1.3. Análisis comparativo de usabilidad

En este análisis comparativo de usabilidad se busca establecer la información relacionada al uso realizado por parte del usuario en las 4 herramientas de Hadoop para el procesamiento de Big Data. En la tabla 19 se describen las características generales que evalúan ventajas y desventajas comparando su usabilidad:

Tabla 19: Análisis comparativo de usabilidad

CARACTERÍSTICAS	**HERRAMIENTAS**			
	Flume	**Hive**	**Sqoop**	**Pig**

Curva de aprendizaje	- Se aprende el uso de la herramienta de entre 2 a 5 días. - La funcionalidad está enfocada a configuraciones para que se deben aplicar para la recolección de la información.	- Se aprende el uso de la herramienta de entre 1 a 5 días. - Hive utiliza el lenguaje de consulta SQL utilizado con los gestores de bases de datos, lo cual lo convierte en un lenguaje más accesible con el que es posible empezar a trabajar desde el principio para cualquier usuario habituado a trabajar con bases de datos.	- Se aprende el uso de la herramienta de entre 1 a 2 días. - Sqoop utiliza drivers JDBC para resolver la conectividad con las bases de datos, es por eso que en los comandos de importación o exportación de datos no son difíciles de aprender y aplicar.	- Se aprende el uso de la herramienta de entre 5 a 7 días. - Pig utiliza su propio lenguaje de alto nivel llamado Pig Latín, el cual es un lenguaje de flujo de datos. Esto implica que el usuario necesita aprender las sentencias que forman parte de Pig Latín para poder utilizar la herramienta.
Nivel de complejidad en la manipulación de datos	- Medio dependiendo de las fuentes de datos para la recolección de la información.	- Bajo. Hive permite la creación de tablas en el repositorio de metadatos de Hive mapeadas hacia la ruta HDFS y explotar la información con HiveQL.	- Bajo. El intercambio de datos entre Hadoop y bases de datos relacionales lo hace basándose en la base de datos de origen para describir el esquema de importación de los datos.	- Alto. Se necesita tener conocimiento de Pig Latín para la codificación de tareas utilizadas en el procesamiento y manejo de datos.
Compatibilidad con distintos sistemas operativos	- Sí. Compatible con Linux y Mac OS.	- Sí. Compatible con Windows, Linux y Mac OS.	- Sí. Compatible con Linux y Mac OS.	- Sí. Compatible con Windows, Linux y Mac OS.
Flexibilidad	- Sí. Posee una arquitectura sencilla y flexible basada en flujos de datos en Streaming.	- Sí. El manejo de grandes volúmenes de datos en el repositorio de metadatos de Hive son altamente flexibles al poder manejarse con lenguaje SQL.	- Sí. Sqoop proporciona una interfaz basada en línea de comandos flexible que se configura para pasar los argumentos necesarios para el manejo de la información.	- Sí. Permite la lectura y manejo de grandes volúmenes de datos (especialmente los no estructurados) mediante su propio lenguaje de alto nivel llamado Pig Latín.

Tiempos de respuesta a nivel de tareas	- No programa tareas. La recolección de datos lo hace bajo demanda.	- No programa tareas. La ejecución de comandos lo hace bajo demanda.	- No programa tareas. Los comandos para manejo de datos son solo bajo demanda.	- Si programa tareas para el procesamiento y manejo de datos.

Fuente: Elaboración propia.

4.3. Beneficios de las herramientas seleccionadas

4.3.1. Beneficios de Flume

Flume tiene los siguientes beneficios:

- Maneja un repositorio central en el HDFS de Hadoop para el almacenamiento de la información. De esa manera la información recolectada por Flume es independiente de otros repositorios de datos.
- Recolecta en tiempo real diferentes fuentes de datos para poder ser analizados con herramientas de Hadoop: Hive, Pig, HBase, entre otros. La información recolectada se la usa para construir modelos de predicción de comportamiento relacionado a varios temas. Esto se convierte en un gran aporte para una empresa y de esa manera sondear como se posiciona en un mercado.
- Al ser parte de Hadoop, su funcionamiento está enfocado al manejo de grandes volúmenes de datos de manera distribuida, confiable y de alta disponibilidad.
- Soporta un gran conjunto de tipos de fuentes y destinos para obtener la información. Apache Flume posee una arquitectura sencilla y flexible basada en flujos de datos en Streaming, principalmente datos estructurados y no estructurados contenidos en archivos de registro, logs, repositorios, registros de datos, servidores web, entre otros.

4.3.2. Beneficios de Hive

Hive tiene los siguientes beneficios:

- Estructura, resume y hace consultas de grandes volúmenes de datos estructurados, semi-estructurados y no estructurados. Muchos de los datos que posee una organización son datos registrados en archivos XML o JSON los cuales procesa Hive para darle una interpretación y uso para la empresa.
- Proporciona una interfaz SQL y un modelo relacional para Hadoop. Un usuario puede hacer uso de la herramienta teniendo conocimientos previos de sentencias del lenguaje SQL. Esto es beneficioso en cuanto al tiempo invertido para explotar las funcionalidades de Hive junto a su curva de aprendizaje.

- Hive es construido bajo un enfoque analítico. Esto significa que Hive no actualiza (update) y elimina (delete) registros específicos en un conjunto de datos, pero es muy eficiente en la lectura y procesamiento de grandes volúmenes de datos, incluso más rápido que las bases de datos relacionales que manejan lenguaje SQL.
- En las organizaciones empresariales que manejan gran volúmenes de datos requieren un análisis rápido de los datos recopilados durante un período de tiempo. Hive es una excelente herramienta para la consulta analítica de datos históricos. Es necesario que los datos estén bien organizados, lo que facilitaría a Hive liberar completamente su procesamiento y su capacidad analítica.

4.3.3. Beneficios de Sqoop

Sqoop tiene los siguientes beneficios:

- Sqoop soporta la importación masiva de datos en el HDFS de Hadoop desde almacenes de datos estructurados como bases de datos relacionales (Teradata, Netezza, Oracle, MySQL, PostgreSQL, etc), almacenes de datos empresariales y sistemas NoSQL (Couchbase). Sqoop posee conectores predeterminados para diferentes bases de datos populares, tales como MySQL, PostgreSQL, Oracle, SQL Server y DB2, pero también incluye un conector JDBC genérico que se usa para conectarse a cualquier base de datos accesible a través de JDBC.
- Sqoop es amigable y funcional con el ecosistema de Hadoop. Su principal beneficio está en la preparación de los datos en el HDFS de Hadoop para que posteriormente puedan ser analizados los datos con otras herramientas de Hadoop: Hive, Pig, HBase, etc.
- Sqoop hace eficiente el análisis de datos. Muchas organizaciones empresariales usan Sqoop para mover grandes volúmenes de datos que no se analizan convencionalmente por bases de datos relacionales, o bien lo hacen pero no cuentan con la programación del Map Reduce propia de Hadoop la cual facilita el procesamiento rápido de Big Data.

4.3.4. Beneficios de Pig

Pig tiene los siguientes beneficios:

- Pig fue diseñado para el entorno de Hadoop en el que existe datos estructurados y no estructurados. Al ser primordial al manejo de grandes volúmenes de datos en una organización empresarial con Pig no se requiere cargar datos en tablas, opera sobre los datos tan pronto como sean copiados en el HDFS de Hadoop.

- El usuario crea funciones en Pig personalizadas para satisfacer sus necesidades particulares de procesamiento de datos. Esto se debe a que Pig Latín proporciona diversos operadores que al utilizarlos permiten desarrollar funciones propias para la lectura, escritura, y procesamiento de grandes volúmenes de datos.
- Al ser fácilmente programable, las tareas complejas que implican transformaciones de datos relacionados entre sí se simplifican y codifican siguiendo una secuencia de flujo de datos. Esto es beneficioso en tiempo, ya que el usuario se centra en la semántica del código y no solo en la ejecución de sus tareas.
- La información procesada y analizada por Pig se la usa para construir modelos de predicción de comportamiento relacionado a varios temas en una organización empresarial. De esta manera se establece gustos, sentimientos, sondeos de un producto, servicio, etc., en el mercado.

4.4. Desarrollo de un prototipo de usabilidad para manejo de las 4 herramientas de Hadoop

Para realizar el prototipo se siguieron los siguientes pasos:

4.4.1. Análisis del prototipo

El prototipo a desarrollar es una aplicación local que se conectará desde el localhost de la máquina local a la terminal del sistema para la ejecución y consulta de comandos de cada una de las 4 herramientas de Hadoop.

Las herramientas utilizadas para el desarrollo son:

- **Servidor JBOSS 7.1.1:** Servidor para publicación y administración de aplicaciones.
- **Eclipse Indigo Service Release 2:** IDE de desarrollo Java.
- **JDK 1.7:** Kit de desarrollo para aplicaciones Java.

El desarrollo de aplicaciones Java bajo el estándar JEE (Java Empresarial) permite la creación de una aplicación escalable, portable y a la vez integrable con tecnologías anteriores. Adicional, el servidor de aplicaciones puede manejar transacciones, la seguridad, concurrencia y gestión de los componentes desplegados. Esto significa que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes, en lugar de tareas de mantenimiento de bajo nivel.

Arquitectura de Aplicaciones JEE utilizada para el prototipo de usabilidad

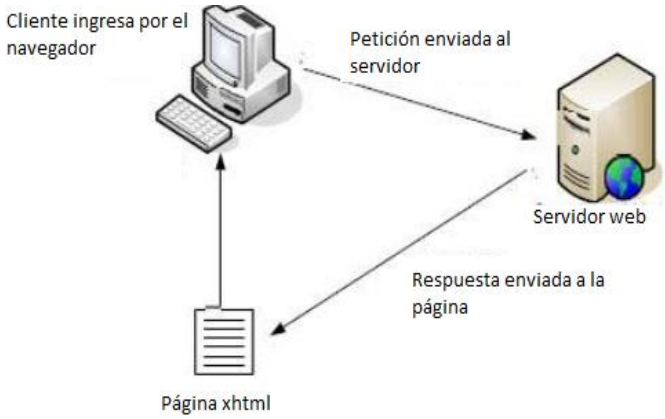


Figura 144: Arquitectura de Aplicaciones JEE
Fuente: Elaboración propia.

Funcionamiento: El usuario ingresa a la aplicación mediante el navegador Web, el cual realiza una petición al servidor de aplicaciones Jboss, el servidor procesa la petición y envía la respuesta al usuario a través de una página xhtml que se visualiza en el navegador.

El prototipo tiene una arquitectura interna por capas y gráficamente se la puede visualizar en la siguiente figura:

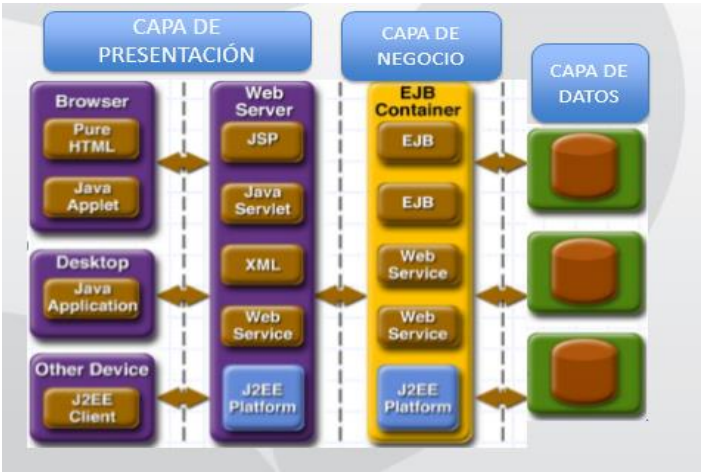


Figura 145: Arquitectura interna por capas del prototipo
Fuente: (Deitel & Deitel, 2013)

El funcionamiento de cada una de las capas es el siguiente:

- **Capa de Presentación:** Capa que contiene la interfaz de usuario de la aplicación.
- **Capa de Negocio:** Capa que contiene los EJB que toman la información de la capa de datos para crear servicios que serán expuestos y usados por la capa superior.
- **Capa de Datos:** Es la capa que se encarga de gestionar el acceso a datos.

4.4.2. Diseño del prototipo

Para el diseño del prototipo es necesario crear un formato o plantilla que contenga las características básicas que permitan a un usuario hacer uso de las 4 herramientas de Hadoop de manera fácil y eficiente.

El formato base del prototipo que debe contener las siguientes opciones:

- Una pantalla de ingreso para el usuario, pero no es necesario que se registre el usuario y contraseña debido a que la ejecución de las herramientas de Hadoop se lo hará con el usuario del sistema y de manera local.
- Una pantalla de opciones para manipular las herramientas de Hadoop. La cual consta de:
    1. **Cabecera:** en la que se encuentre el logo de la UTPL, el nombre del prototipo y un botón de salida del sistema.
    2. **Menú general:** en la que se pueda seleccionar las herramientas: Hadoop, Flume, Hive, Sqoop, Pig y navegación en el HDFS.
    3. **Botones de ejecución:** que permitan ejecutar las herramientas seleccionadas en el menú general.
    4. **Menú de ayuda para el usuario:** en la que se ubica sentencias y comandos de ayuda al usuario para utilizarlos en cada una de las herramientas seleccionadas en el menú general.
    5. **Terminal del sistema:** que será utilizada para ejecutar comandos y sentencias de las herramientas de Hadoop.

De manera general, el diseño del prototipo en plantilla es el siguiente:

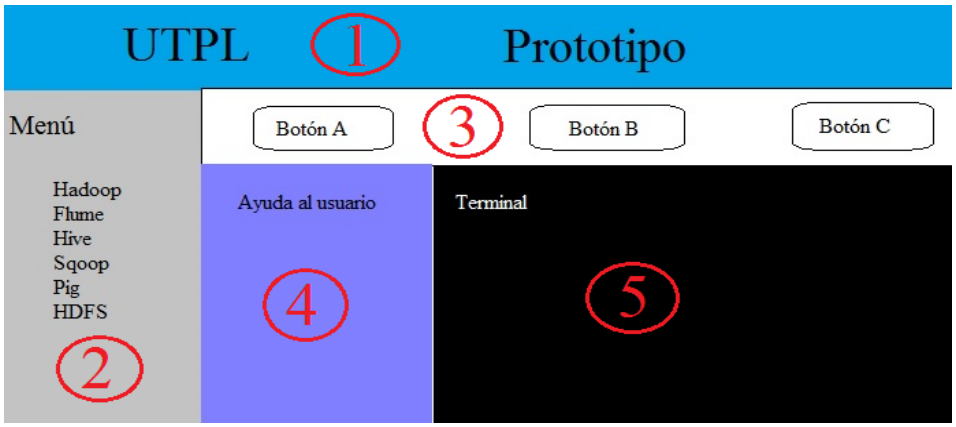


Figura 146: Diseño del prototipo base
Fuente: Elaboración propia.

Siguiendo el prototipo base, el diseño de las páginas a utilizar en el prototipo final son:

Objeto	Descripción
login.xhtml	Permite el ingreso al sistema. 

Figura 147: Pantalla de ingreso al sistema
Fuente: Elaboración propia.

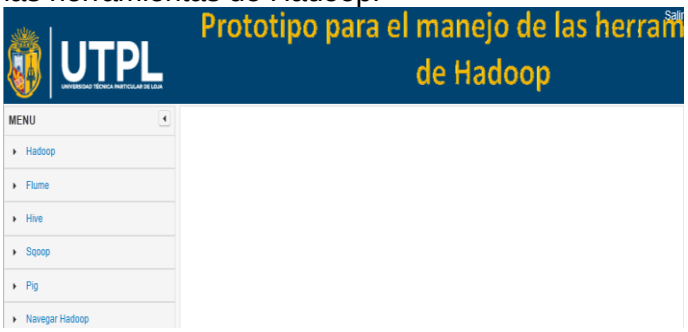
Objeto	Descripción
home.xhtml	Pantalla que contiene el menú para el manejo de las herramientas de Hadoop. 

Figura 148: Pantalla con menú general
Fuente: Elaboración propia.

Objeto	Descripción
hadoop.xhtml	Pantalla que permite iniciar y detener el servicio de Hadoop. 

Figura 149: Pantalla para Hadoop
Fuente: Elaboración propia.

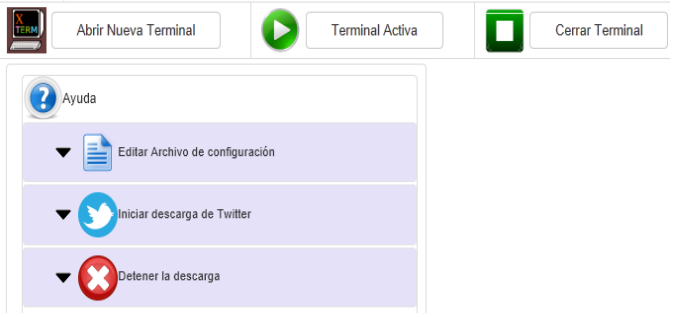
Objeto	Descripción
flume.xhtml	Pantalla que muestra ayuda al usuario sobre la ejecución de Flume, además permite abrir una terminal para el uso y manipulación de la herramienta. 

Figura 150: Pantalla para Flume
Fuente: Elaboración propia.

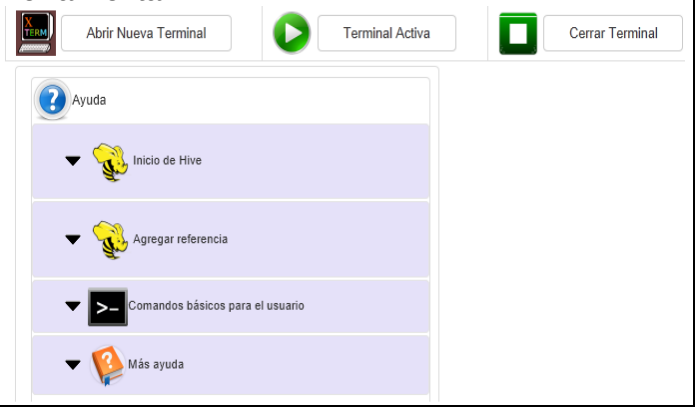
Objeto	Descripción
Hive.xhtml	Pantalla que muestra ayuda al usuario sobre la ejecución de Hive, además permite abrir una terminal para el uso y manipulación de la herramienta. 

Figura 151: Pantalla para Hive
Fuente: Elaboración propia.

Objeto	Descripción
sqoop.xhtml	Pantalla que muestra ayuda al usuario sobre la ejecución de Sqoop, además permite abrir una terminal para el uso y manipulación de la herramienta.

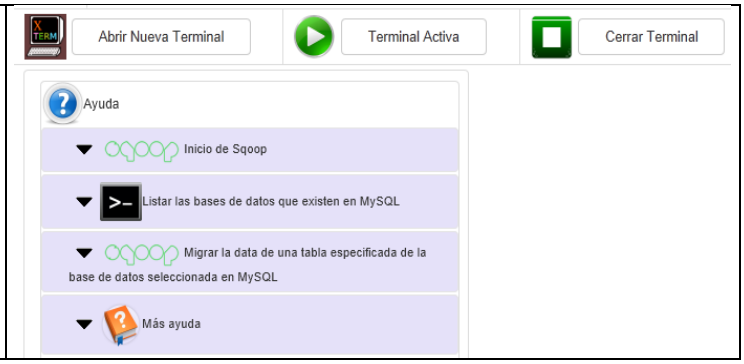


Figura 152: Pantalla para Sqoop
Fuente: Elaboración propia.

Objeto	Descripción
pig.xhtml	Pantalla que muestra ayuda al usuario sobre la ejecución de Pig, además permite abrir una terminal para el uso y manipulación de la herramienta.

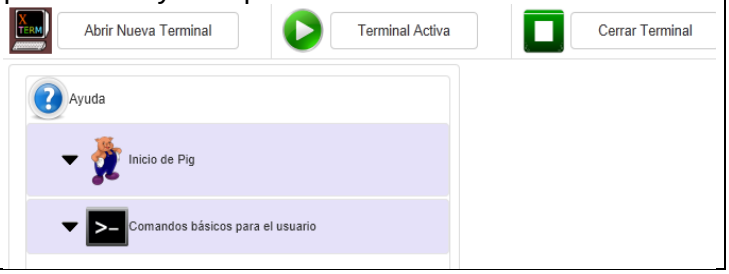


Figura 153: Pantalla para Pig
Fuente: Elaboración propia.

Objeto	Descripción
navegarHadoop.xhtml	Pantalla que muestra ayuda al usuario sobre la navegación de Hadoop, además permite abrir una terminal para el uso y manipulación del HDFS de Hadoop.

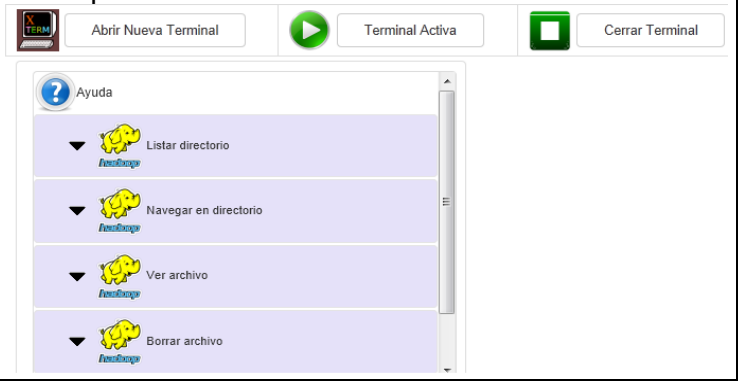


Figura 154: Pantalla para navegar en Hadoop
Fuente: Elaboración propia.

4.4.3. Implementación del prototipo

Como se muestra en la figura 154, el prototipo de usabilidad se compone de tres proyectos:

- **Prototipo:** Es un proyecto EAR que contiene el archivo comprimido de la aplicación, el cual se desplegará en el servidor de aplicaciones.
- **Prototipo EJB:** Proyecto de tipo EJB que contiene la capa de negocio de la aplicación, es decir los servicios para realizar la conexión a la herramienta Hadoop. Este proyecto genera un jar que puede ser utilizado en cualquiera de los otros proyectos.
- **PrototipoWeb:** Es un proyecto de tipo Web que contiene la capa de presentación de la aplicación, es decir todas las páginas que conforman la aplicación, este proyecto genera un archivo WAR.

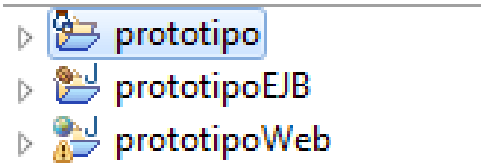


Figura 155: Proyecto general
Fuente: Elaboración propia.

Prototipo EJB:

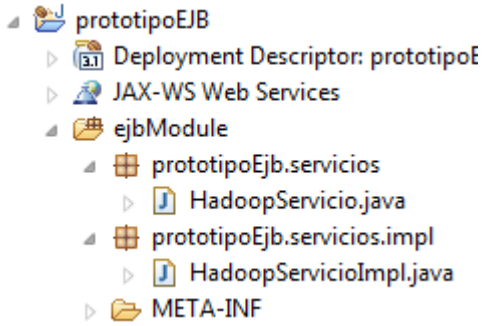


Figura 156: Prototipo EJB
Fuente: Elaboración propia.

El prototipo EJB contiene dos paquetes:

- **PrototipoEjb.servicios:** Interfaz para acceder al servicio de conexión con Hadoop.

```
@Local
public interface HadoopServicio {

    Connection conectar(String usuario,String passw,String host);

    void desconectar(Connection conn,Session sess);
}
```

Figura 157: PrototipoEjb.servicios
Fuente: Elaboración propia.

- **PrototipoEjb.servicios.impl:** Implementación de la interfaz para la conexión con Hadoop.

```
@Stateless
public class HadoopServicioImpl implements HadoopServicio {

    @Override
    public Connection conectar(String usuario,String passw,String serverIp)
    {
            Connection conn=null;
        try {
            conn = new Connection(serverIp);
            conn.connect();
            boolean isAuthenticated = conn.authenticateWithPassword(
                    usuario, passw);
            if (isAuthenticated == false)
                throw new IOException("Authentication failed.");
                    } catch (IOException e) {
            e.printStackTrace(System.err);

        }
        return conn;

    }
```

Figura 158: PrototipoEjb.servicios.impl
Fuente: Elaboración propia.

Prototipo Web:

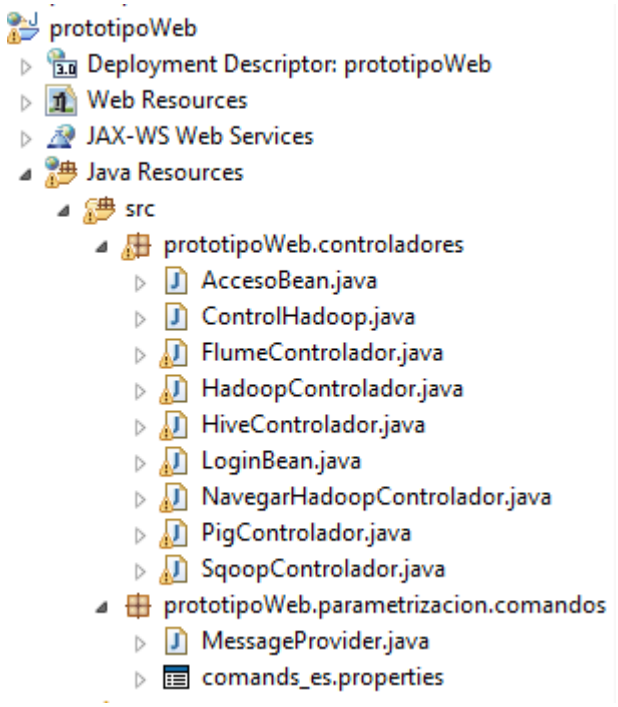


Figura 159: Prototipo Web
Fuente: Elaboración propia.

El prototipo de usabilidad contiene dos paquetes prototipoWeb.controladores y prototipoWeb.parametrizacion.comandos, los mismos que se describen a continuación**:**

- **prototipoWeb.controladores:** Clases Java que permiten la interacción entre las páginas Web y los métodos de negocio de la aplicación.

  LoginBean: contiene los métodos de la página para el ingreso al sistema, se inicializa las banderas para el control y manejo de las terminales abiertas en la aplicación.

```
public class LoginBean {

    private boolean conectado;

    private boolean iniciadoHadoop;

    private boolean abiertoTerminal;

    @PostConstruct
    public void init() {

        iniciadoHadoop=false;
        abiertoTerminal=false;
    }

    public boolean isConectado() {
        return conectado;
    }
```

Figura 160: LoginBean
Fuente: Elaboración propia.

HadoopControlador: Contiene los métodos de inicio y cierre de la terminal para el manejo de las herramienta de Hadoop, para esto se utiliza la API de java Runtime que permite la ejecución de comandos en el sistema operativo local. Los métodos más importantes son iniciarRuntime() que permite mostrar en pantalla un emulador de la terminal y detenerRuntime() para cerrar y quitar de pantalla el emulador de la terminal.

```
public String iniciarRuntime()
{
     String s="";
     String lineaFinal="";
     String comando = MessageProvider.getGeneralMessage("iniciarHadoop");

     String[] cmd = { "/bin/sh", "-c", comando };
        Process p;
        try {
            p = Runtime.getRuntime().exec(cmd);
            BufferedReader br = new BufferedReader(
                new InputStreamReader(p.getInputStream()));

            while ((s = br.readLine()) != null)
            {
                System.out.println("line: " + s);
                lineaFinal=s;
            }
            p.waitFor();
            if (!lineaFinal.equals("")) {
                if (lineaFinal
                        .equals("localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-diego-VirtualBox.out")) {
                    loginBean.setIniciadoHadoop(true);
                    textoSalida = "Servicio iniciado correctamente.";
                } else {
                    loginBean.setIniciadoHadoop(true);
                    String[] cadena = lineaFinal.split("\\.");
                    if (cadena[cadena.length - 1].equals(" Stop it first")) {
                        textoSalida = "Servicio iniciado correctamente.";
                    }
                }

            } else {
                loginBean.setIniciadoHadoop(false);
                textoSalida = "Problemas al iniciar servicio.";
            }


            System.out.println ("exit: " + p.exitValue());
            p.destroy();
        } catch (Exception e) {
```

Figura 161: HadoopControlador para iniciar
Fuente: Elaboración propia.

```
public String detenerRuntime()
{
     String s="";
     String lineaFinal="";
     String comando = MessageProvider.getGeneralMessage("detenerHadoop");
     String[] cmd = { "/bin/sh", "-c", comando };
        Process p;
        if (!loginBean.isIniciadoHadoop()) {
            textoSalida = "El servicio no se encuentra iniciado.";
        } else {
        try {
            p = Runtime.getRuntime().exec(cmd);
            BufferedReader br = new BufferedReader(
                new InputStreamReader(p.getInputStream()));

            while ((s = br.readLine()) != null)
            {
                System.out.println("line: " + s);
                lineaFinal=s;
            }
            p.waitFor();

                if (!lineaFinal.equals("")) {
                    if (lineaFinal.equals("no proxyserver to stop")) {
                        loginBean.setIniciadoHadoop(false);
                        textoSalida = "Servicio detenido correctamente.";
                    } else {
                        textoSalida = "Servicio no detenido.";
                    }
                } else {
                    textoSalida = "Problemas al detener servicio.";
                }

            System.out.println ("exit: " + p.exitValue());
            p.destroy();

        } catch (Exception e) {

        }
        }
        return lineaFinal;
}
```

Figura 162: HadoopControlador para detener
Fuente: Elaboración propia.

FlumeControlador: Contiene los métodos de inicio y cierre de la terminal para la manipulación de la herramienta Flume.

HiveControlador: Contiene los métodos de inicio y cierre de la terminal para la manipulación de la herramienta Hive.

SqoopControlador: Contiene los métodos de inicio y cierre de la terminal para la manipulación de la herramienta Sqoop.

PigControlador: Contiene los métodos de inicio y cierre de la terminal para la manipulación de la herramienta Pig.

NavegarHadoopControlador: Contiene los métodos de inicio y cierre de la terminal para la navegación en Hadoop.

En los controladores que se usan para el manejo de las herramientas de Hadoop, los métodos más importantes son iniciarRuntime() (figura 162) el cual que permite mostrar en pantalla un emulador de la terminal, recuperarRuntime() (figura 163) el cual recupera

el emulador de la terminal que se está utilizando y detenerRuntime() (figura 164) utilizado para cerrar y quitar de pantalla el emulador de la terminal.

```
public String iniciarRuntime()
{
     String s="";
     String lineaFinal="";
     if(loginBean.isIniciadoHadoop()){
     if(loginBean.isAbiertoTerminal())
     {
         FacesContext context=FacesContext.getCurrentInstance();
         context.addMessage(null, new FacesMessage("Error", "Existe una terminal activa por favor dar clic en cerrar terminal"));

     }
     else
     {
     String comando = MessageProvider.getGeneralMessage("abrirTerminal");

        Process p;
        try {
            Runtime rt = Runtime.getRuntime();
            Process pr = rt.exec(comando);
            //BufferedReader br = new BufferedReader(
             //   new InputStreamReader(p.getInputStream()));
            loginBean.setAbiertoTerminal(true);
                        System.out.println ("exit: " + pr.exitValue());

                        pr.destroy();
        } catch (Exception e) {

        }
     }
     }
     else
     {
         FacesContext context=FacesContext.getCurrentInstance();
         context.addMessage(null, new FacesMessage("Error", "El servicio Hadoop no se encuentra iniciado"));
     }
        return lineaFinal;
}
```

Figura 163: Método iniciarRuntime()
Fuente: Elaboración propia.

```
public String recuperarRuntime()
{
     String s="";
     String lineaFinal="";
     if(loginBean.isIniciadoHadoop()){

     String comando = MessageProvider.getGeneralMessage("recuperarTerminal");

        Process p;
        try {
            Runtime rt = Runtime.getRuntime();
            Process pr = rt.exec(comando);
            //BufferedReader br = new BufferedReader(
             //   new InputStreamReader(p.getInputStream()));
            loginBean.setAbiertoTerminal(true);
                        System.out.println ("exit: " + pr.exitValue());

                        pr.destroy();
        } catch (Exception e) {

        }
     }

     else
     {
         FacesContext context=FacesContext.getCurrentInstance();
         context.addMessage(null, new FacesMessage("Error", "El servicio Hadoop no se encuentra iniciado"));
     }
        return lineaFinal;
}
```

Figura 164: Método recuperarRuntime()
Fuente: Elaboración propia.

```
public String detenerRuntime()
{
     String s="";
     String lineaFinal="";

     String comando = MessageProvider.getGeneralMessage("abrirTerminal");

        Process p;
        try {
            Runtime rt = Runtime.getRuntime();
            Process pr = rt.exec("/usr/bin/killall xterm");
            //BufferedReader br = new BufferedReader(
             //   new InputStreamReader(p.getInputStream()));
            loginBean.setAbiertoTerminal(false);
                            System.out.println ("Term exit: " + pr.exitValue());

                            pr.destroy();
        } catch (Exception e) {


     }
        return lineaFinal;
}
```

Figura 165: Método detenerRuntime()
Fuente: Elaboración propia.

- **prototipoWeb.parametrizacion.comandos:** Contiene el archivo de propiedades con todos los comandos a utilizarse y también una clase Java que permite la obtención de cada uno de estos comandos.

name	value
iniciarHadoop	cd /usr/local/hadoop/sbin; ./start-all.sh
abrirTerminal	/usr/bin/xterm -geometry 100x25+900+300
recuperarTerminal	wmctrl -a hduser@diego-VirtualBox: ~/Desktop/jboss-as-7.1.1.Final/bin
detenerHadoop	cd /usr/local/hadoop/sbin; ./stop-all.sh
userFlume	hduser
passwUserFlume	password123
iniciarDescarga	cd $FLUME_HOME; bin/flume-ng agent -n TwitterAgent --conf ./conf/ -f conf/flume-twitter....

Figura 166: Comandos prototipoWeb.parametrizacion.comandos
Fuente: Elaboración propia.

```
public class MessageProvider {

    private static final String VAR_GENERAL_MESSAGES_BUNDLE = "msg";
    private static final ResourceBundle GENERAL_BUNDLE;


    static {
        GENERAL_BUNDLE = getBundle(VAR_GENERAL_MESSAGES_BUNDLE);

    }

    private static ResourceBundle getBundle(String varResourceBoundle) {
        FacesContext context = FacesContext.getCurrentInstance();
        return context.getApplication().getResourceBundle(context,
                varResourceBoundle);
    }

    private static String getValue(String key, ResourceBundle bundle) {

        String result = null;
        try {
            result = bundle.getString(key);
        } catch (MissingResourceException e) {
            result = "¿¿¿" + key + "??? ";
        }
        return result;
```

Figura 167: Clase prototipoWeb.parametrizacion.comandos
Fuente: Elaboración propia.

Carpeta pages: contiene todas las páginas Web de la aplicación.

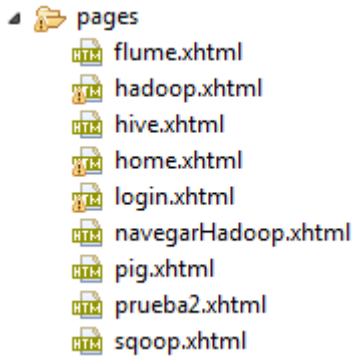


Figura 168: Carpeta pages
Fuente: Elaboración propia.

Carpeta resources: posee todos los recursos usados en la aplicación: imágenes, hojas de estilos, templates, archivo Java Script usado para abrir los tabs de la aplicación.

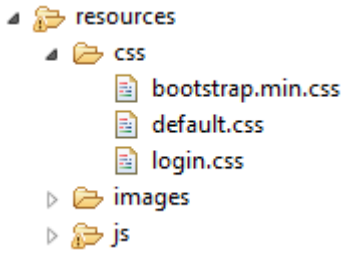


Figura 169: Carpeta resources
Fuente: Elaboración propia.

Prototipo:

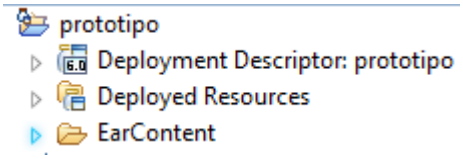


Figura 170: Prototipo EAR
Fuente: Elaboración propia.

Contiene el archivo comprimido que debe ser copiado en la carpeta deployments del servidor de aplicaciones para ser desplegado, cuando el archivo se ha desplegado correctamente, se crea un archivo con el mismo nombre con la extensión .deployed.

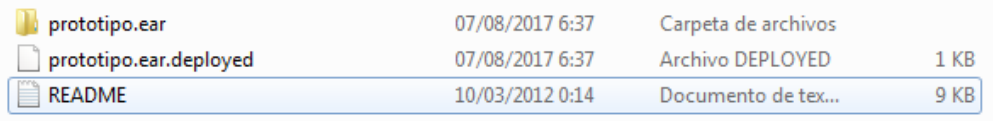


Figura 171: Carpeta prototipo EAR
Fuente: Elaboración propia.

4.4.4. Ejecución del prototipo

El archivo comprimido con el resultado final del prototipo, debe ser copiado en la carpeta deployments del servidor de aplicaciones para ser desplegado. A continuación se debe levantar el servidor de aplicaciones JBoss para la ejecución de la aplicación. Esto se hace ejecutando el comando **./stanalone.sh –b 0.0.0.0** en la terminal del sistema, ubicándonos en la carpeta "bin" del directorio en el que se encuentra el Servidor JBoss 7.1.1.

Como referencia se tiene el siguiente ejemplo ejecutado en la máquina local:

```
hduser@diego-VirtualBox:~$ cd Desktop/jboss-as-7.1.1.Final/bin
hduser@diego-VirtualBox:~/Desktop/jboss-as-7.1.1.Final/bin$ ./standalone.sh -b 0.0.0.0
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: /home/hduser/Desktop/jboss-as-7.1.1.Final

  JAVA: /usr/lib/jvm/java-7-openjdk-amd64/bin/java

  JAVA_OPTS:  -server -XX:+UseCompressedOops -XX:+TieredCompilation -Xms64m -Xmx512m -X
e -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djbo
fault.config=standalone.xml
```

Figura 172: Levantar servidor JBoss
Fuente: Elaboración propia.

A continuación, para desplegar el prototipo se debe escribir la siguiente URL en la barra de direcciones del navegador: http://localhost:8080/prototipoWeb/pages/login.xhtml y se mostrará la siguiente pantalla:

Figura 173: Prototipo en navegador
Fuente: Elaboración propia.

4.4.5. Plan de pruebas del prototipo

En este punto se realizaron consultas sobre el prototipo de usabilidad para comparar que los resultados arrojados durante la primera y segunda vuelta electoral 2017 sean iguales a los que se generaron al utilizar las herramientas individualmente y sin ayuda para el usuario.

4.4.5.1. Resultados en el prototipo del caso de estudio 1.

Configuración de Flume en caso de estudio 1.

Al realizar la descarga de datos en el prototipo de usabilidad mediante la ejecución del agente Flume se obtuvo el siguiente resultado:

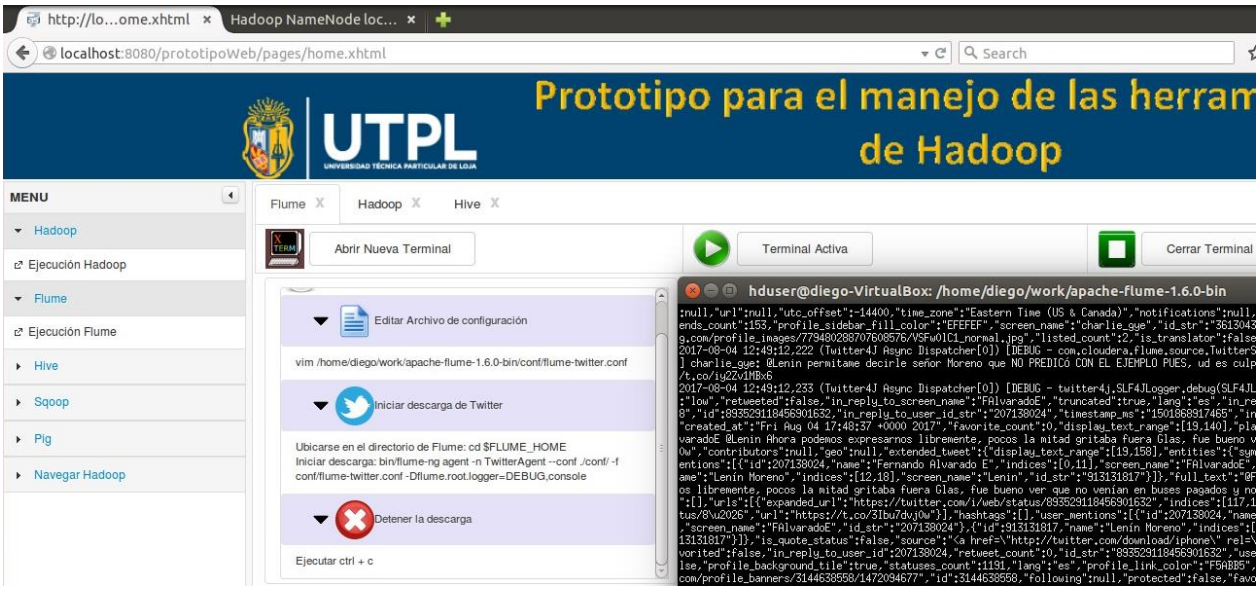


Figura 174: Ejecución del agente Flume en prototipo
Fuente: Elaboración propia.

El resultado de la figura 173 es igual al del agente Flume sin el uso del prototipo (apartado 3.3.1.2), esto se demuestra en la figura 174 y de esta manera queda probado que Flume funciona de manera correcta en el prototipo desarrollado.

Figura 175: Comparación de resultado de Agente Flume en 1ra vuelta
Fuente: Elaboración propia.

Aplicación de los comandos y sintaxis de Hive en caso de estudio 1.

Para confirmar el funcionamiento correcto de Hive en el prototipo, se ejecutará los comandos de consulta sobre los datos generados durante la primera vuelta electoral 2017.

- **Total de registros en 1ra vuelta**

Al ejecutar en el prototipo de usabilidad el comando para contabilizar el total de registros en la primera vuelta electoral se obtuvo el siguiente resultado:

Figura 176: Resultado del total de registros durante la 1ra vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 175 es igual al de Hive sin el uso del prototipo (apartado 3.3.2.2), esto se demuestra en la figura 176.

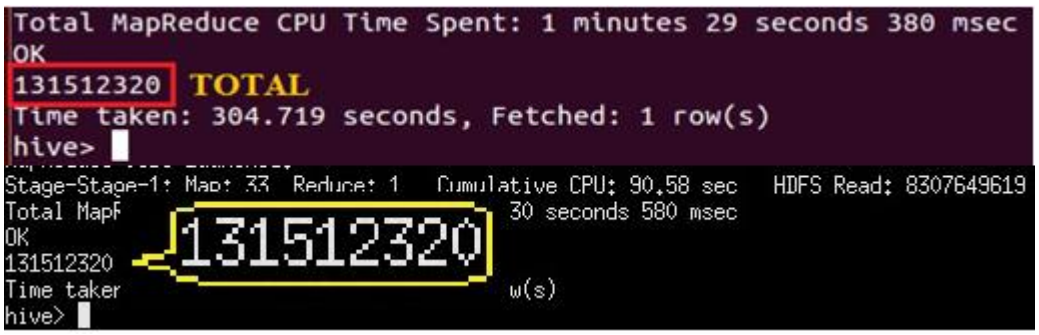


Figura 177: Comparación de total de registros en 1ra vuelta
Fuente: Elaboración propia.

- **Número de veces que se mencionaron a los candidatos en 1ra vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar el número de veces que fueron mencionados los candidatos en la 1ra vuelta se obtuvo el siguiente resultado:

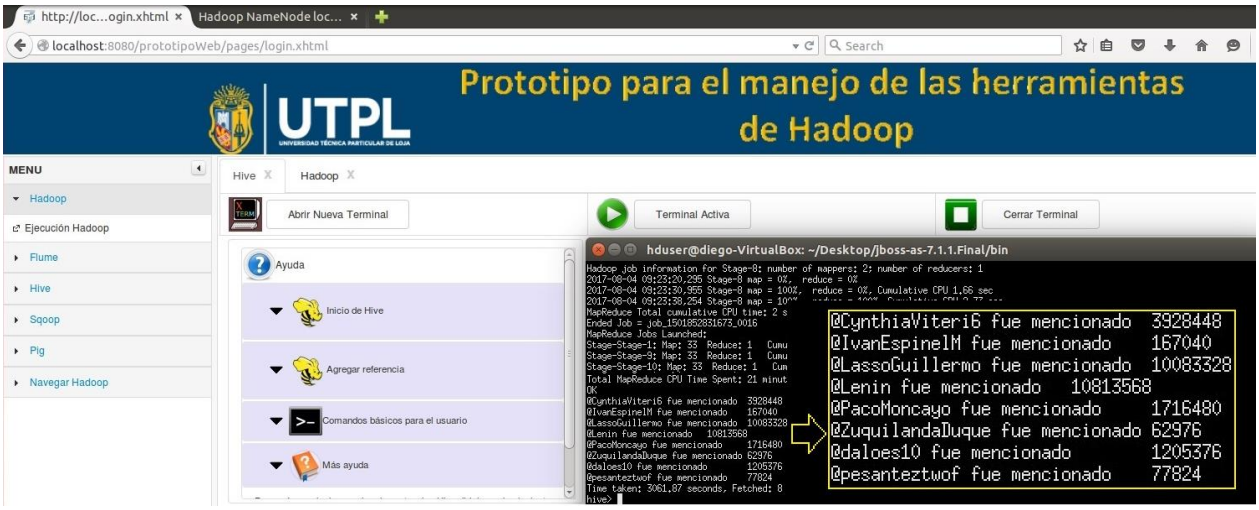


Figura 178: Resultado número de veces que se mencionaron a los candidatos en la 1ra vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 177 es igual al de Hive sin el uso del prototipo (apartado 3.3.2.2), esto se demuestra en la figura 178.

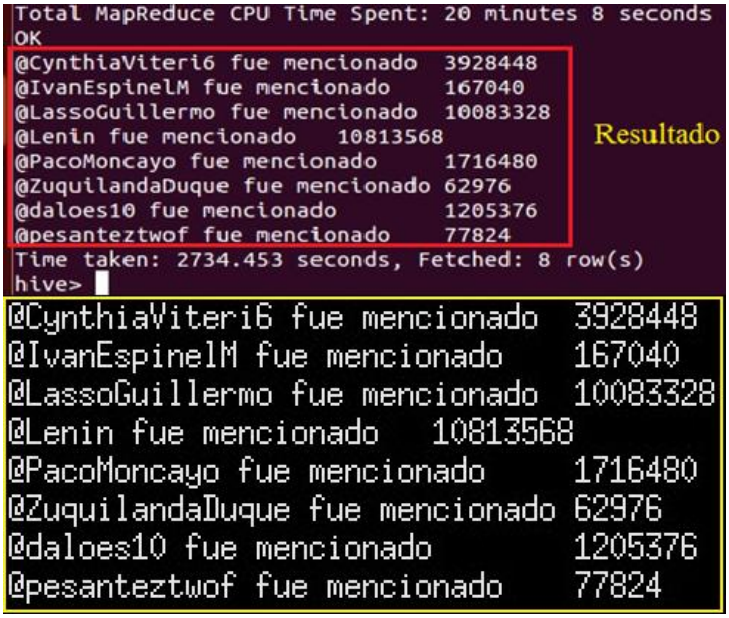


Figura 179: Comparación de número de veces que se mencionaron a los candidatos en 1ra vuelta
Fuente: Elaboración propia.

- **Cantidad de tweets que publicó cada candidato en 1ra vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de tweets que publicó cada candidato en la 1ra vuelta se obtuvo el siguiente resultado:

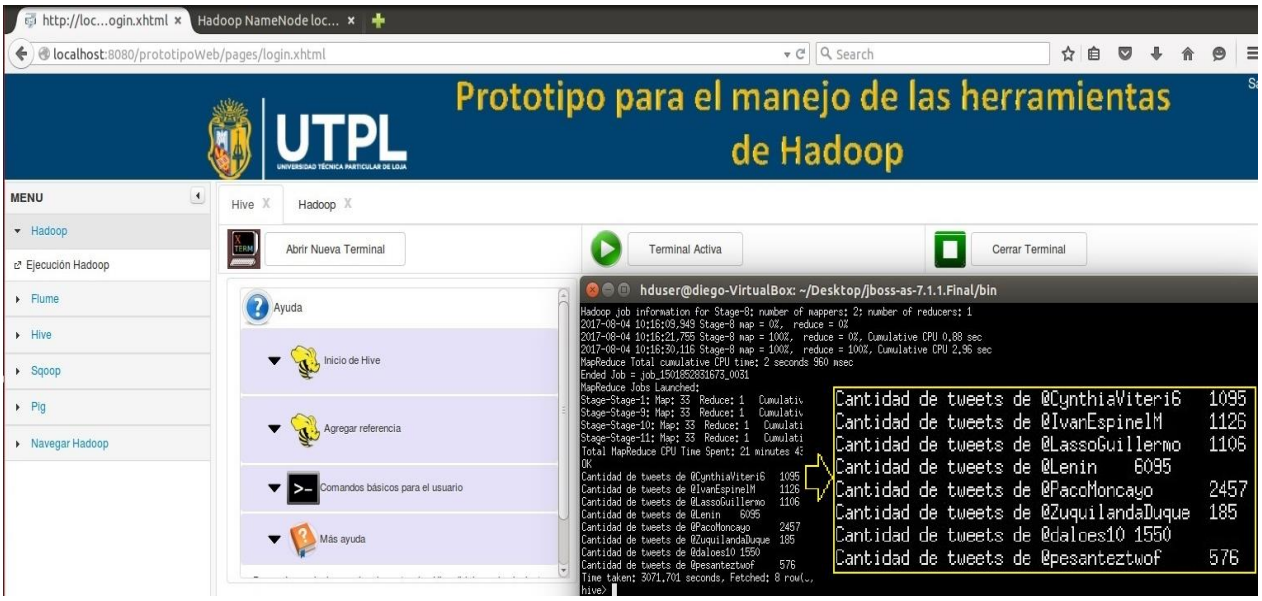


Figura 180: Resultado de cantidad de tweets que publicó cada candidato en 1ra vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 179 es igual al de Hive sin el uso del prototipo (apartado 3.3.2.2), esto se demuestra en la figura 180.

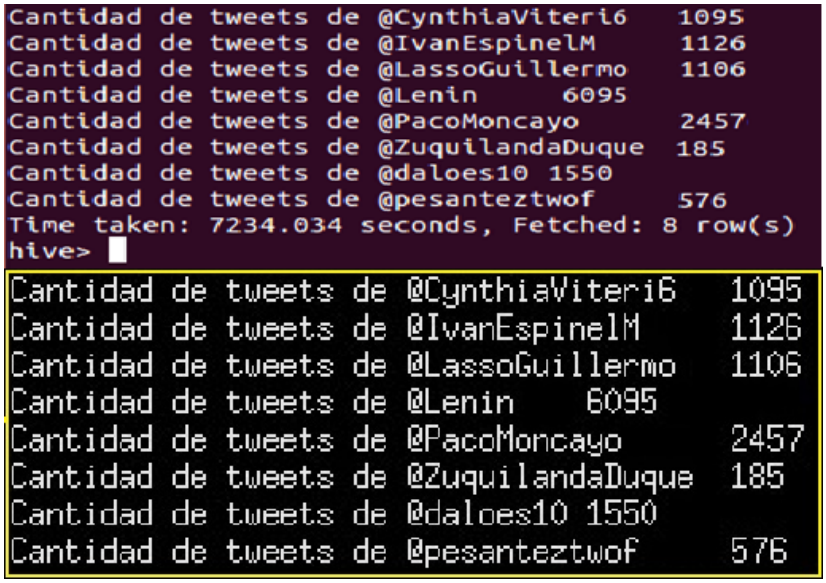


Figura 181: Comparación de cantidad de tweets que publicó cada candidato en 1ra vuelta
Fuente: Elaboración propia.

- **Tweets más retweeteados (texto y mayor a 200000 veces) en 1ra vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de tweets más retweeteados (texto y mayor a 200000 veces) en 1ra vuelta se obtuvo el siguiente resultado:

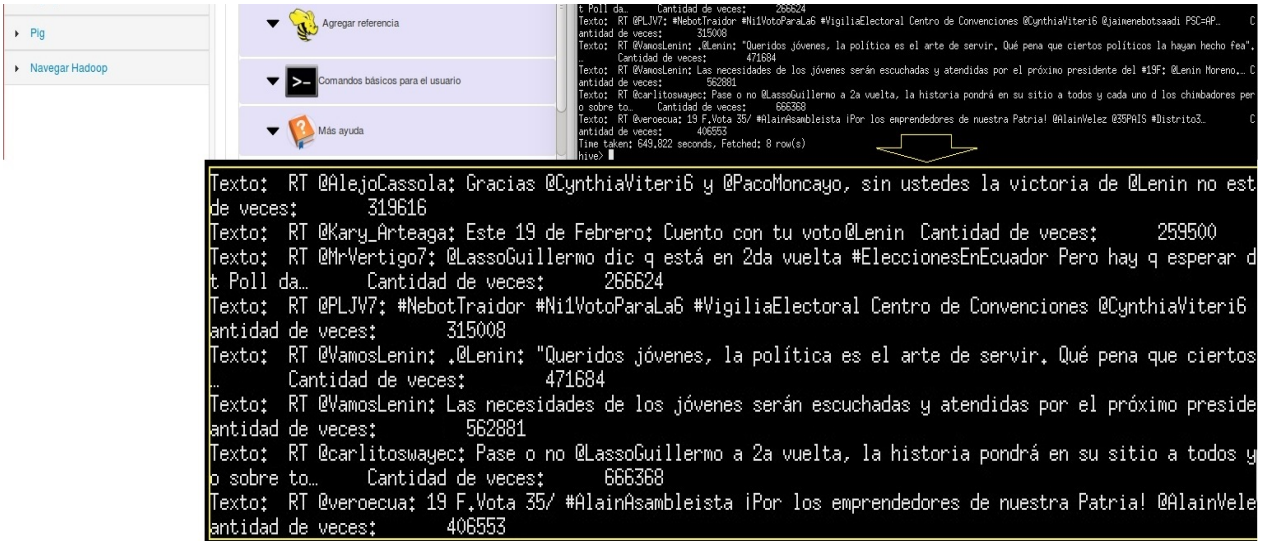


Figura 182: Resultado de tweets más retweeteados (texto y mayor a 200000 veces) en 1ra vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 181 es igual al de Hive sin el uso del prototipo (apartado 3.3.2.2), esto se demuestra en la figura 182.

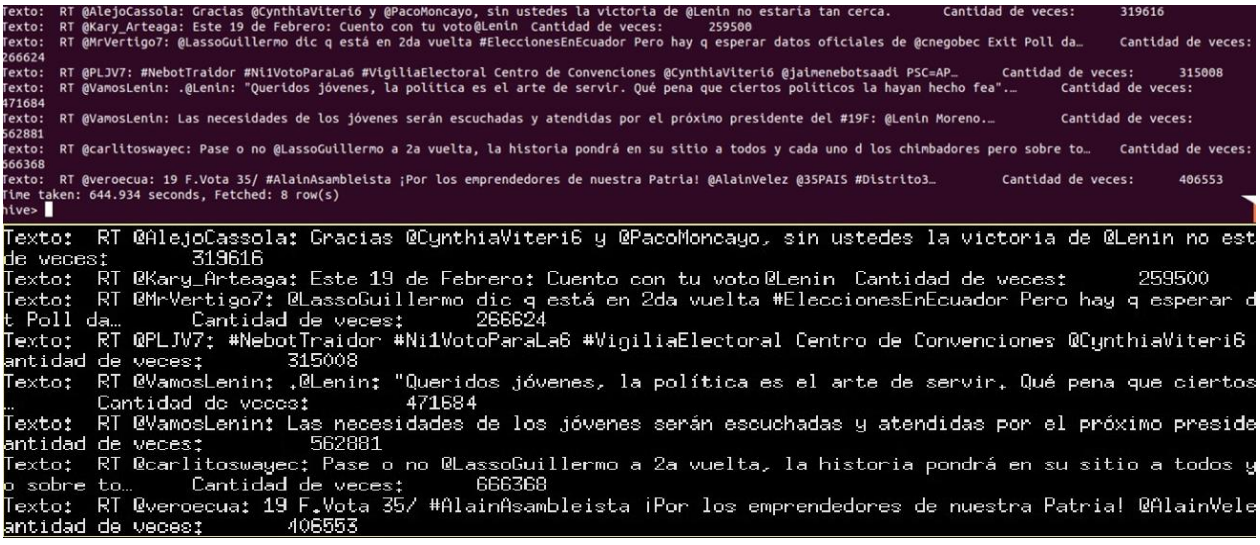


Figura 183: Comparación de tweets más retweeteados (texto y mayor a 200000 veces) en 1ra vuelta
Fuente: Elaboración propia.

- **Cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta se obtuvo el siguiente resultado:

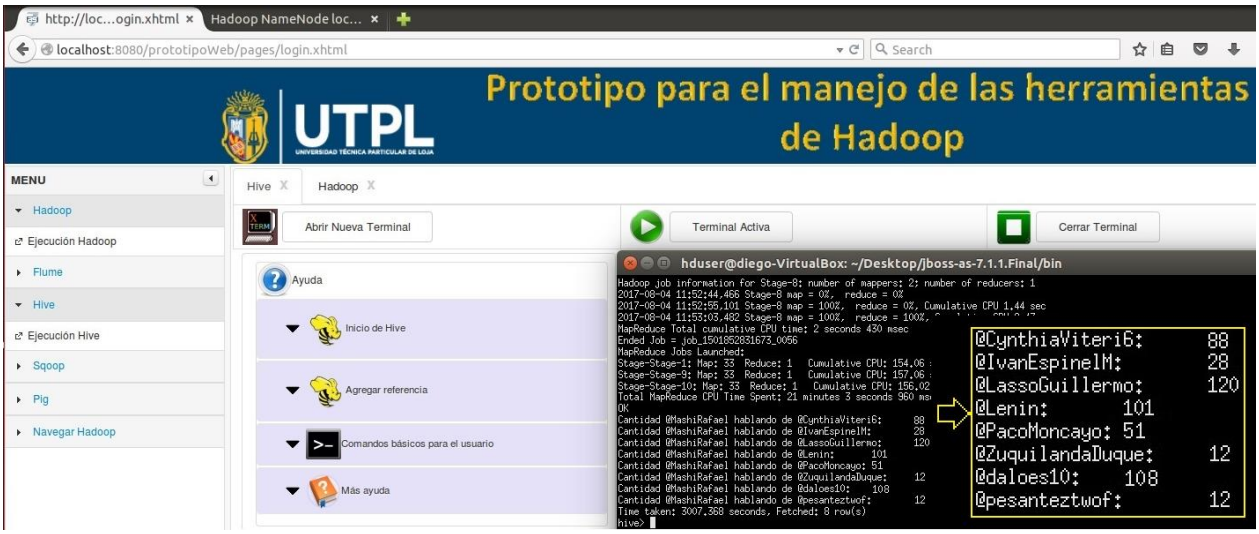


Figura 184: Resultado de cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 183 es igual al de Hive sin el uso del prototipo (apartado 3.3.2.2), esto se demuestra en la figura 184.

Figura 185: Comparación cantidad de tweets de Rafael Correa hablando de candidatos en 1ra vuelta
Fuente: Elaboración propia.

De manera general, se ha confirmado que los datos generados en Hive al usar el prototipo son iguales y confiables que los datos generados al utilizar las herramientas individualmente y sin ayuda para el usuario, de esta manera queda demostrado que Hive funciona de manera correcta en el prototipo desarrollado.

4.4.5.2. Resultados en el prototipo del caso de estudio 2.

Aplicación de los comandos y sintaxis de Sqoop en caso de estudio 2.

Al realizar la descarga de datos de MySQL al HDFS de Hadoop utilizando la ejecución de Sqoop se obtuvo el siguiente resultado:

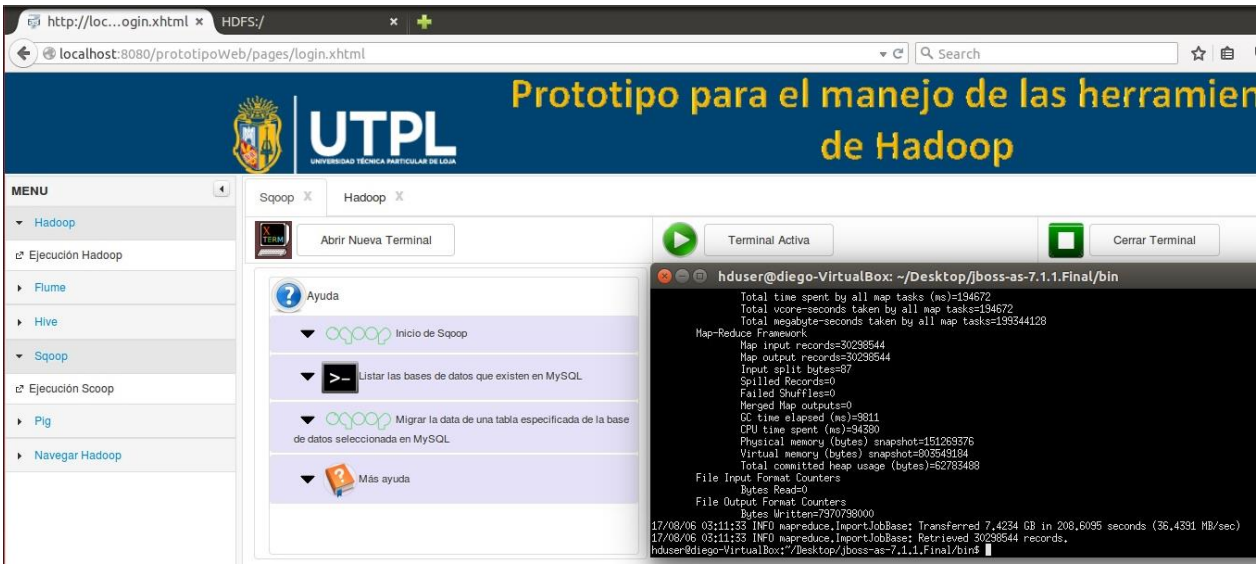


Figura 186: Resultado de descargar datos de MySQL a HDFS usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 185 es igual a la descarga de Sqoop sin el uso del prototipo (apartado 3.3.3.2), esto se demuestra en la figura 186 y de esta manera queda probado que Sqoop funciona de manera correcta en el prototipo desarrollado

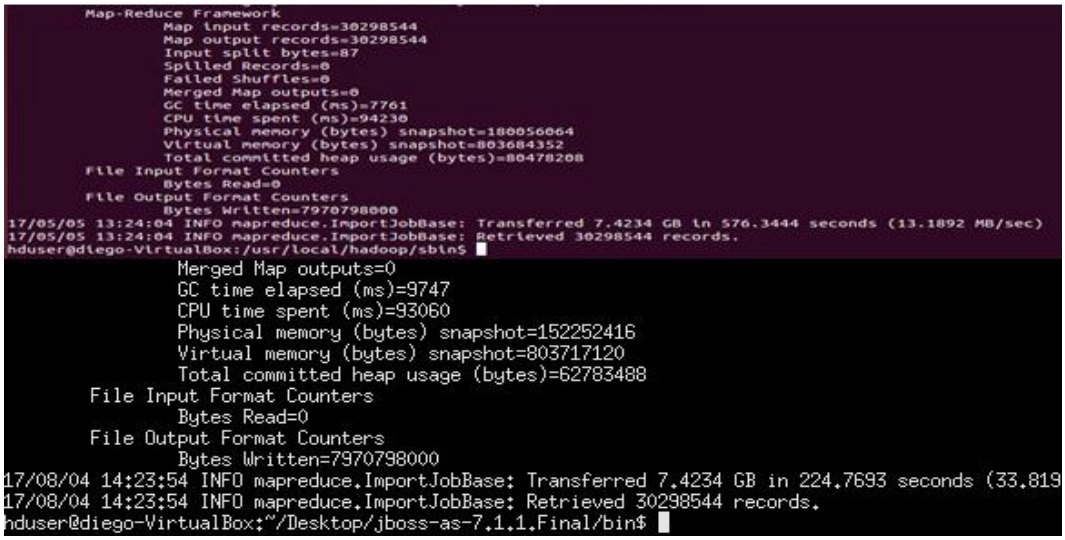


Figura 187: Comparación de descarga de datos de MySQL a HDFS mediante Sqoop
Fuente: Elaboración propia.

Aplicación de los comandos y sintaxis de Pig en caso de estudio 2.

- **Total de registros en 2da vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar el total de registros en la 2da vuelta se obtuvo el siguiente resultado:

Figura 188: Resultado de total de registros en 2da vuelta utilizando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 187 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 188.

Figura 189: Comparación de total de registros en 2da vuelta
Fuente: Elaboración propia.

- **Cantidad de veces que se mencionaron a los candidatos en 2da vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de veces que se mencionó a @Lenin en la 2da vuelta se obtuvo el siguiente resultado:

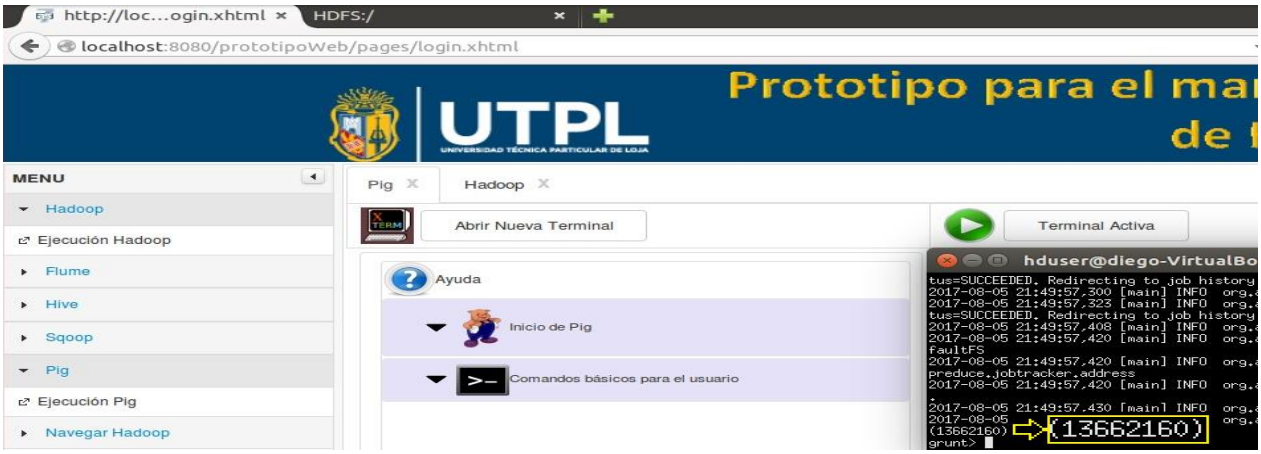


Figura 190: Resultado de cantidad de veces que se mencionó a @Lenin en la 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 189 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 190.

Figura 191: Comparación de cantidad de veces que se mencionó a @Lenin en la 2da vuelta
Fuente: Elaboración propia.

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de veces que se mencionó a @LassoGuillermo en la 2da vuelta se obtuvo el siguiente resultado:

Figura 192: Resultado de cantidad de veces que se mencionó a @LassoGuillermo en la 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 191 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 192.

Figura 193: Comparación de cantidad de veces que se mencionó a @LassoGuillermo en la 2da vuelta
Fuente: Elaboración propia.

- **Cantidad de veces mencionando Rafael Correa a los candidatos en 2da vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de veces que mencionó Rafael Correa a los candidatos en la 2da vuelta se obtuvo el siguiente resultado:

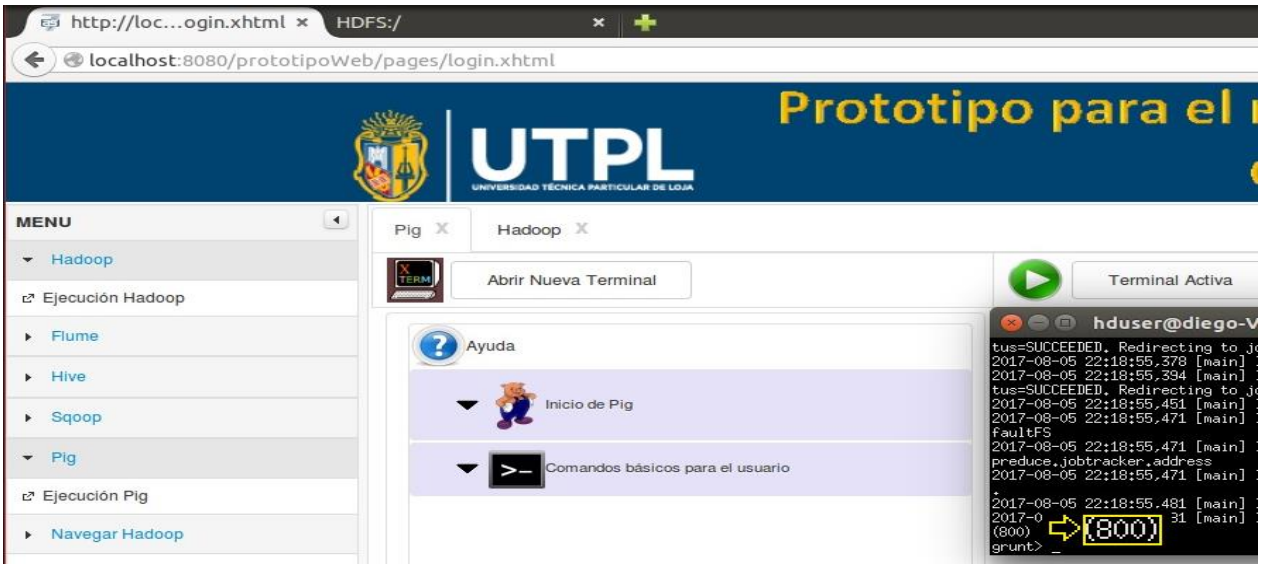


Figura 194: Resultado del total de registros publicados por @MashiRafael en Pig con prototipo
Fuente: Elaboración propia.

El resultado de la figura 193 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 194.

Figura 195: Comparación de cantidad de veces mencionando Rafael Correa a los candidatos en 2da vuelta
Fuente: Elaboración propia.

- **Análisis de sentimientos hacia @Lenin en 2da vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos positivos hacia @Lenin en la 2da vuelta se obtuvo el siguiente resultado:

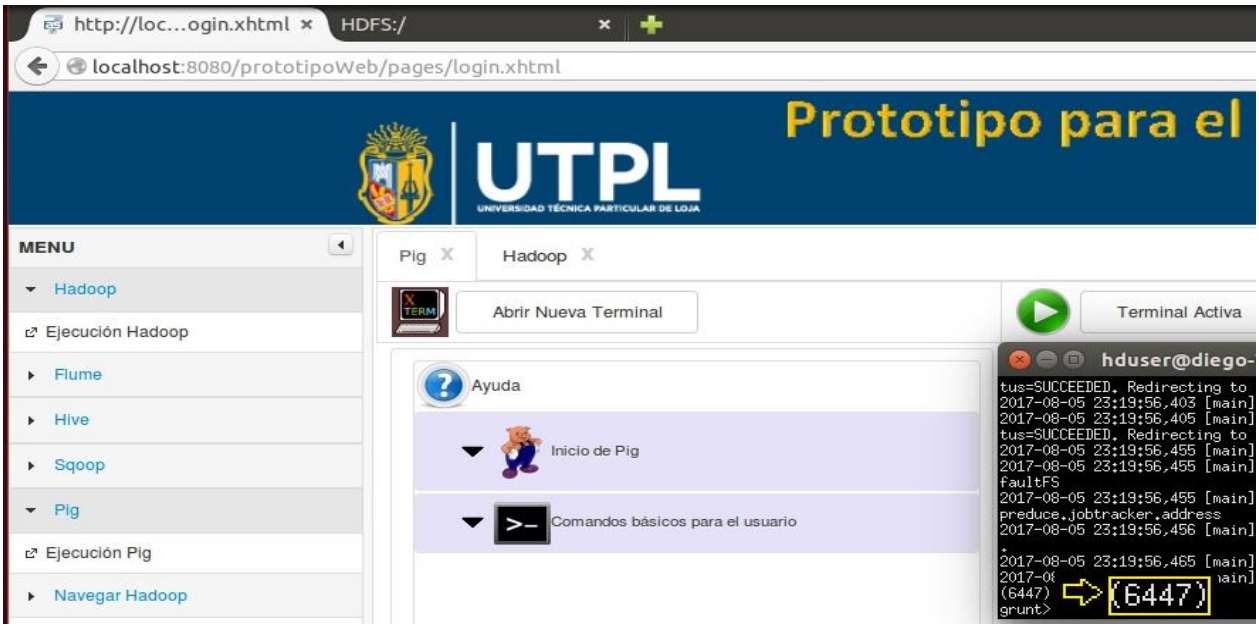


Figura 196: Resultado del total de sentimientos positivos hacia @Lenin en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 195 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 196.

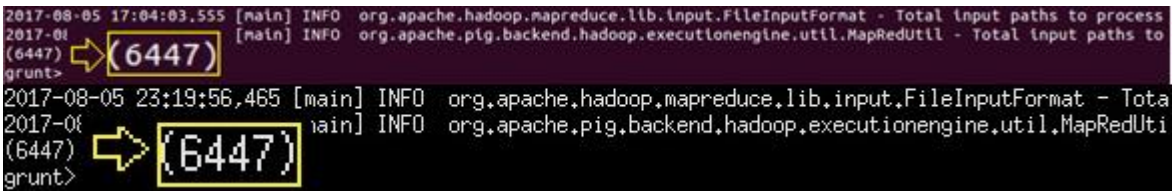


Figura 197: Comparación de sentimientos positivos hacia @Lenin en 2da vuelta
Fuente: Elaboración propia.

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos negativos hacia @Lenin en la 2da vuelta se obtuvo el siguiente resultado:

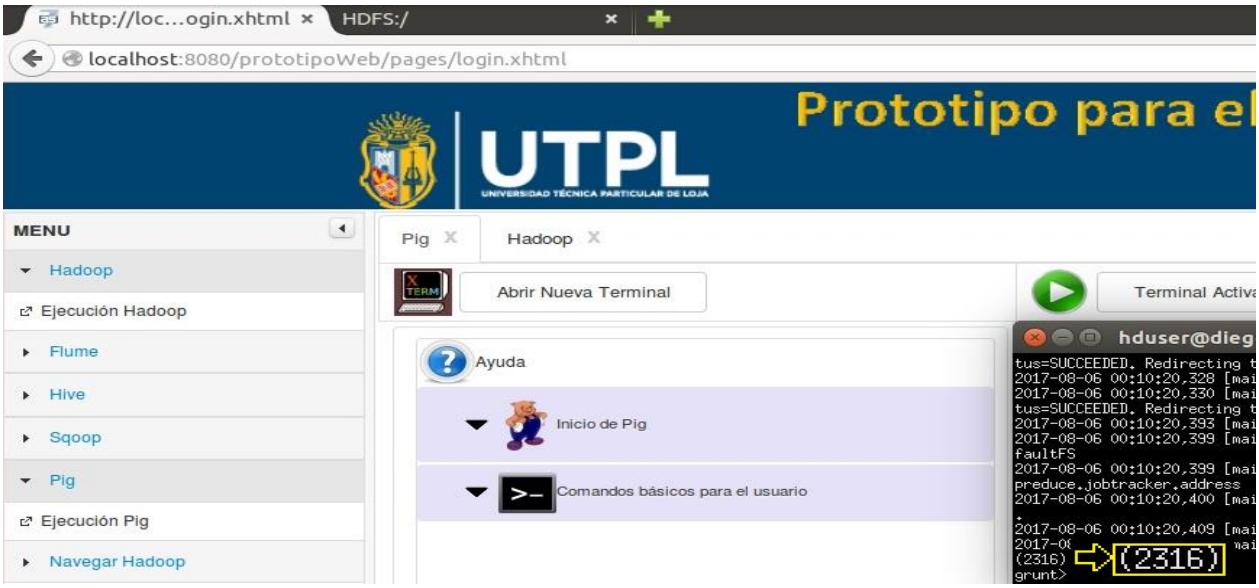


Figura 198: Resultado del total de sentimientos negativos hacia @Lenin en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 197 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 198.

Figura 199: Comparación de sentimientos negativos hacia @Lenin en 2da vuelta
Fuente: Elaboración propia.

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos neutrales hacia @Lenin en la 2da vuelta se obtuvo el siguiente resultado:

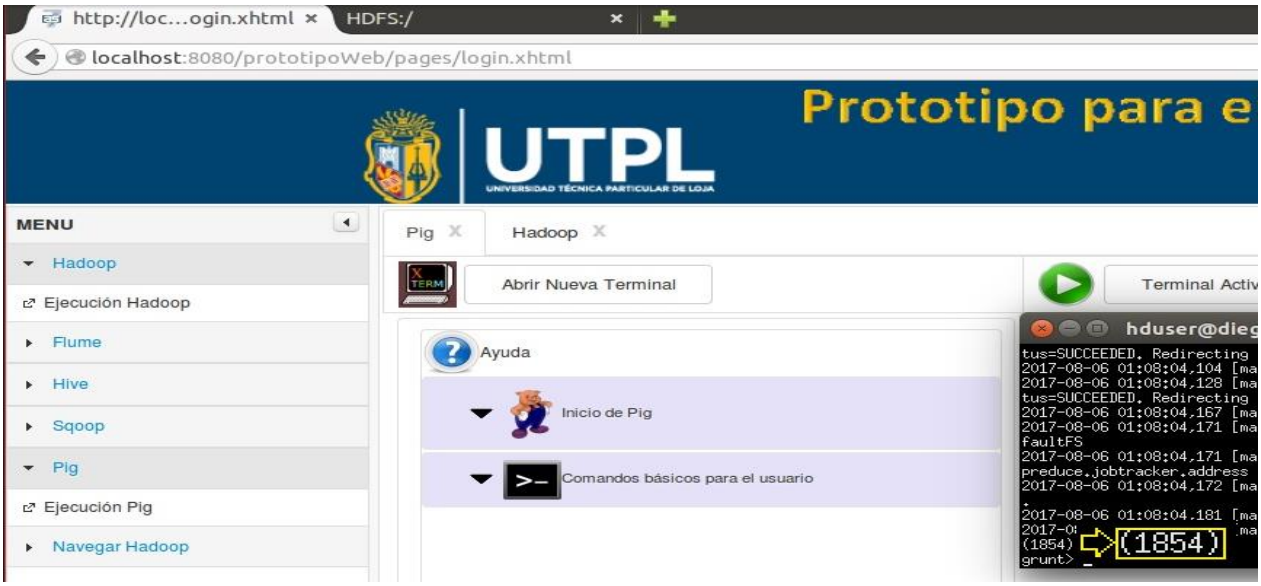


Figura 200: Resultado del total de sentimientos neutrales hacia @Lenin en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 199 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 200.

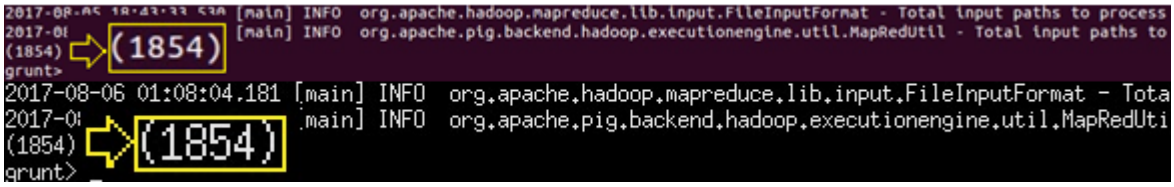


Figura 201: Comparación de sentimientos neutrales hacia @Lenin en 2da vuelta
Fuente: Elaboración propia.

- **Análisis de sentimientos hacia @LassoGuillermo en 2da vuelta**

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos positivos hacia @LassoGuillermo en la 2da vuelta se obtuvo el siguiente resultado:

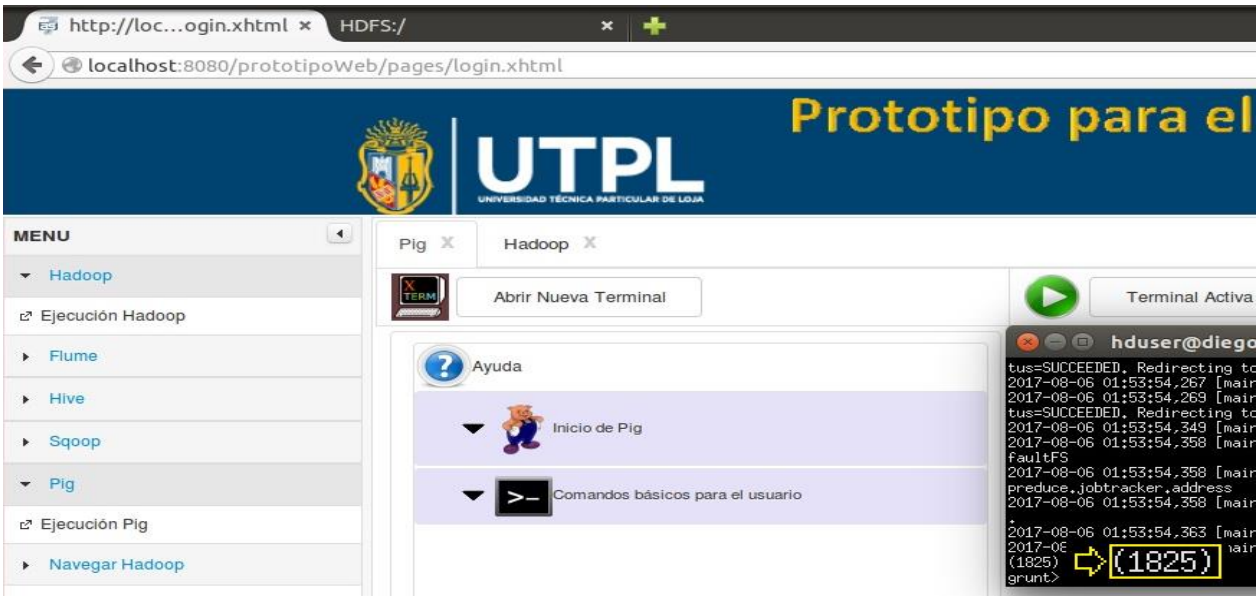


Figura 202: Resultado del total de sentimientos positivos hacia @LassoGuillermo en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 201 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 202.

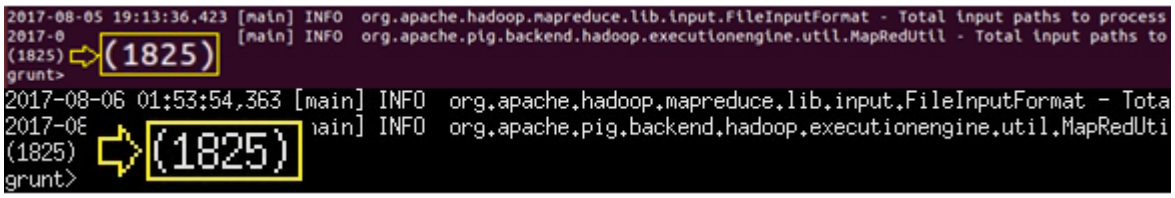


Figura 203: Comparación de sentimientos positivos hacia @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos negativos hacia @LassoGuillermo en la 2da vuelta se obtuvo el siguiente resultado:

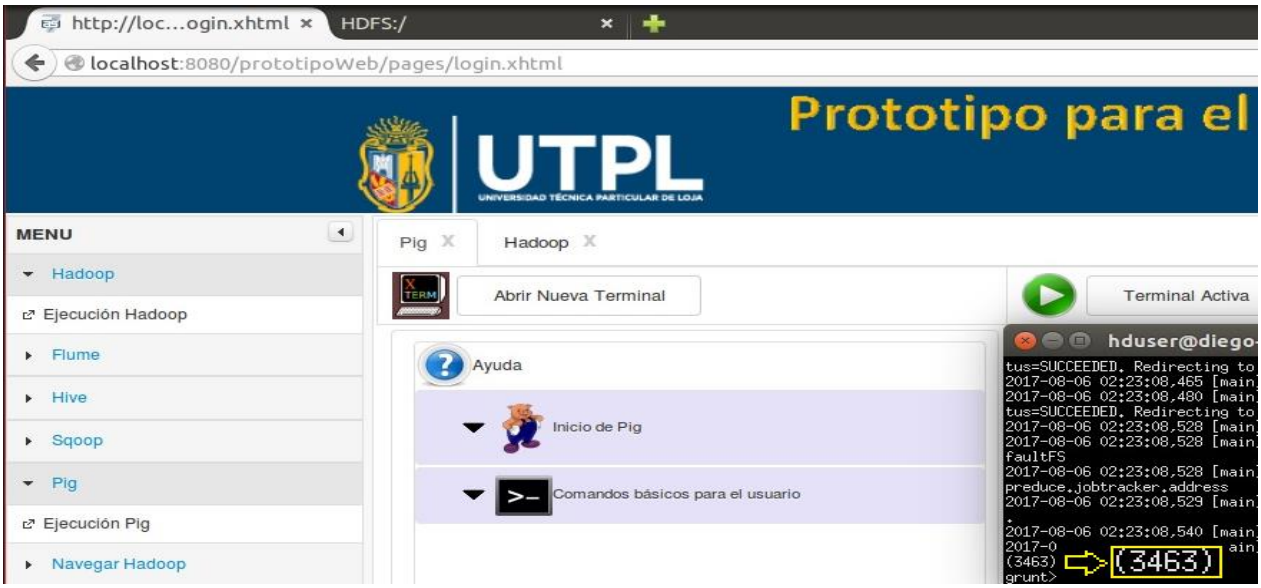


Figura 204: Resultado del total de sentimientos negativos hacia @LassoGuillermo en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 203 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 204.

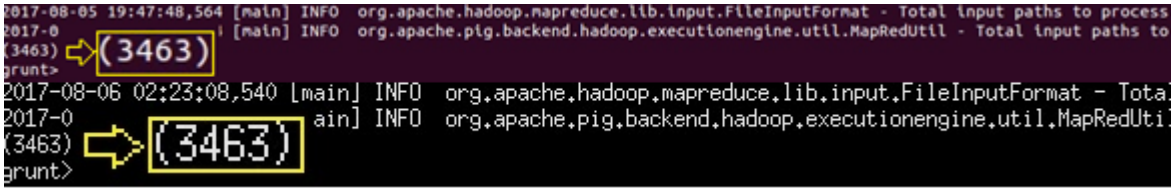


Figura 205: Comparación de sentimientos negativos hacia @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

Al ejecutar sobre el prototipo de usabilidad el comando para contabilizar la cantidad de sentimientos neutrales hacia @LassoGuillermo en la 2da vuelta se obtuvo el siguiente resultado:

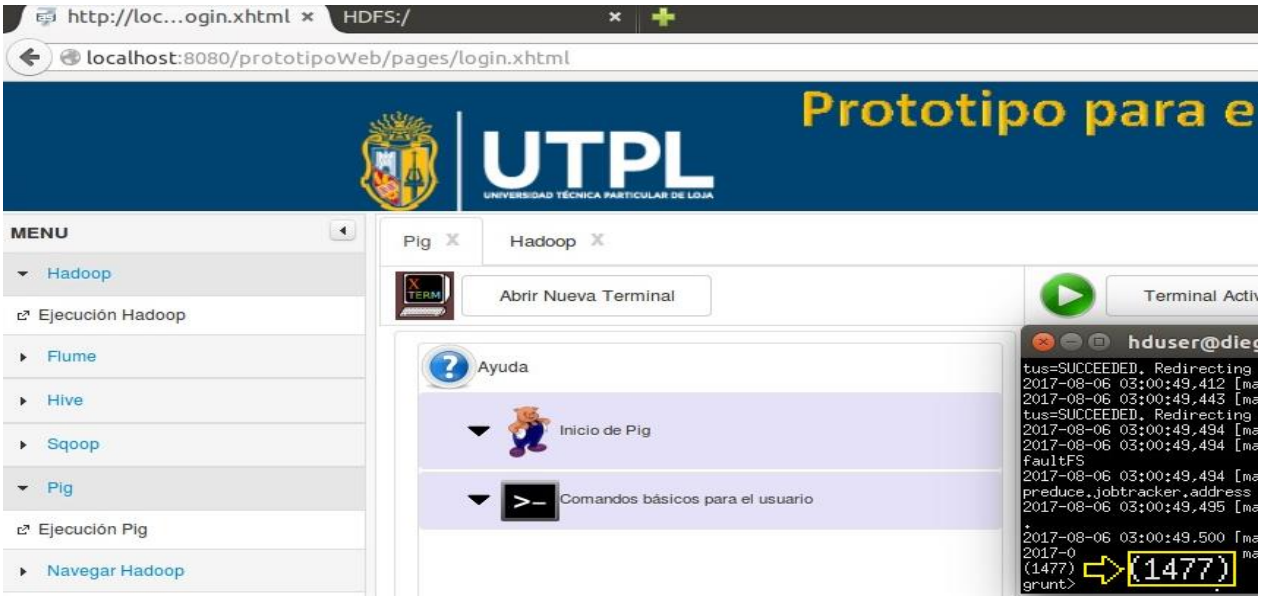


Figura 206: Resultado del total de sentimientos neutrales hacia @LassoGuillermo en 2da vuelta usando el prototipo
Fuente: Elaboración propia.

El resultado de la figura 205 es igual al de Pig sin el uso del prototipo (apartado 3.3.4.2), esto se demuestra en la figura 206.

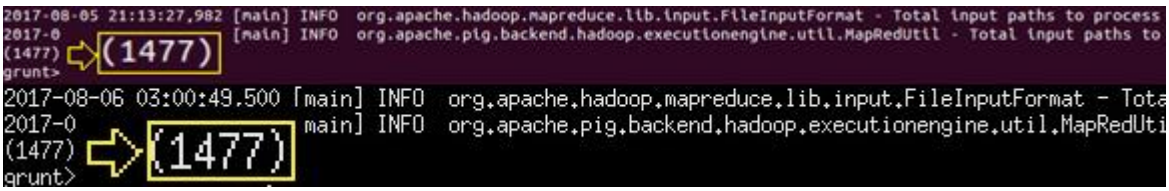


Figura 207: Comparación de sentimientos neutrales hacia @LassoGuillermo en 2da vuelta
Fuente: Elaboración propia.

De manera general, se ha confirmado que los datos generados en Pig al usar el prototipo son iguales y confiables que los datos generados al utilizar las herramientas individualmente y sin ayuda para el usuario, de esta manera queda demostrado que Pig funciona de manera correcta en el prototipo desarrollado.

CONCLUSIONES

En base a las definiciones y resultados obtenidos del análisis de las herramientas podemos concluir lo siguiente:

- Al hablar de Big Data, existen muchas definiciones y conceptos que se encuentran relacionadas de alguna o de otra manera por términos que poseen similares terminologías. Se concluye que el término de Big Data hace relación al análisis, procesamiento y almacenamiento de grandes volúmenes de datos que se originan de diferentes fuentes de información conformando un conjunto de datos estructurados y no estructurados que generan un valor para la toma de decisiones.
- Mucha de la información que se genera constantemente se la obtiene de diferentes fuentes de datos (las redes sociales, correos electrónicos, internet, fotos, clips de videos, sensores, etc.) que por su volumen de crecimiento exponencial es denominada como Big Data. Esta variedad de datos están clasificados como estructurados, semi-estructurados y no estructurados la cual es considerada por los expertos como una fuente de información aprovechable, analizable y lista para ser explotada.
- Constantemente se generan y liberan nuevos proyectos o herramientas de tecnología que forman parte del ecosistema de Hadoop. Estas herramientas se han convertido en la parte fundamental de muchos proyectos y sistemas en las organizaciones, ya que se integran entre ellas y complementan su funcionamiento para el manejo de grandes volúmenes de datos de manera rápida y eficiente.
- Realizando una investigación a nivel corporativo se ha establecido que en organizaciones empresariales que manejan Big Data las principales herramientas del ecosistema de Hadoop que son utilizadas son: Flume, Hive, Sqoop y Pig. Estas herramientas dan una contribución significativa a una organización, aportando valor para que la toma de decisiones sean más inteligentes, rápidas y que marquen la diferencia.
- Al realizar un análisis comparativo general de las herramientas seleccionadas de Hadoop, la herramienta que presenta mayor facilidad para la instalación y configuración es Pig. Las propiedades que se deben aplicar en Hadoop para su funcionamiento son fáciles de configurar, al contrario de la herramienta Hive. Hive tiene un grado alto de dificultad en su instalación y configuración, ya que utiliza Derby para el almacenamiento de sus metadatos. Las propiedades de configuración de Derby en Hadoop son varias y al no estar correctamente implementadas lanzarán una excepción. Es necesario resaltar que las 4 herramientas de Hadoop seleccionadas tienen licencia de software libre, están

enfocadas al trabajo con Big Data y se integran de manera exitosa con otras herramientas del ecosistema de Hadoop.

- Al comparar la funcionalidad de las herramientas seleccionadas de Hadoop, Hive es la herramienta que nos da una mejor funcionalidad para el análisis y procesamiento masivo de datos, debido a que permite el manejo y utilización de datos almacenados en Hadoop a través del lenguaje tipo SQL denominado HiveQL. Este lenguaje de consulta SQL es utilizado en los gestores de bases de datos tradicionales, lo cual lo convierte en un lenguaje accesible para su manejo por parte de cualquier usuario habituado a manejar bases de datos. De las 4 herramientas de Hadoop seleccionadas es Pig la única herramienta que permitió manejar concurrencia en la máquina local para ejecutar diferentes programas MapReduce independientemente una de otra.
- Al analizar la usabilidad de las herramientas seleccionada de Hadoop, Sqoop es la herramienta que mejor usabilidad presentó para el manejo de grandes volúmenes de datos. Su curva de aprendizaje es de 2 días, ya que utiliza drivers JDBC para resolver la conectividad con las bases de datos, es por eso que sus comandos de importación o exportación de datos son fáciles de aprender y aplicar. Sin embargo, en Sqoop los tiempos de respuesta a nivel de tareas depende de la ejecución de comandos por parte del usuario, al igual que en Flume y Hive. En este punto, Pig sobresale de las otras 3 herramientas, ya que se programan tareas para el procesamiento y manejo de datos gracias a su propio lenguaje de alto nivel llamado Pig Latín. Algo que se tiene que resaltar, es que las 4 herramientas de Hadoop seleccionadas presentan compatibilidad a más de 1 sistema operativo y poseen flexibilidad al manejo de varios tipos de datos.
- El diseño y desarrollo de un prototipo de usabilidad en el que se maneja las 4 herramientas seleccionadas de Hadoop (apartado 4.4), mejora de gran manera el procesamiento masivo de datos que forman parte de Big Data. Su funcionalidad permite al usuario levantar de manera rápida el núcleo de procesamiento de Hadoop, brindando también un menú intuitivo y fácil de entender para cada una de sus herramientas.
- La utilización del prototipo de usabilidad ha permitido comparar de manera eficiente los datos estructurados y no estructurados que han sido generados durante la primera y segunda vuelta electoral 2017. Los resultados obtenidos son iguales tanto al utilizar las herramientas de forma individual sin el prototipo, así como también haciendo uso del prototipo desarrollado.

RECOMENDACIONES

- Para iniciar con la obtención y procesamiento de Big Data, es recomendable definir un modelo de negocio que tenga una clara justificación, motivación y los objetivos por el cual se quiere iniciar el proceso de extracción de Big Data. No todos los problemas de negocio que implique el manejo de grandes volúmenes de datos se lo hace con un procesamiento de Big Data. Además, recordar que el problema de negocio debe tener una o más características de Big Data, tales como: volumen, velocidad, variedad, veracidad, valor.
- Para el usuario que vaya a realizar una tarea de procesamiento masivo de datos en el ordenador y si desea que los datos cargados en el sistema se almacenen en tablas locales, se recomienda que utilice Hive, ya que su manipulación es similar al mismo tipo de sentencias del lenguaje SQL.
- En el caso de necesitar que el procesamiento de grandes volúmenes de información sean más rápidos para generar resultados, se recomienda usar un mejor procesador con mayor memoria, ya que la eficiencia del procesamiento masivo de datos mejora con un ordenador de mejores características tecnológicas.
- La calidad de los datos es uno de los grandes obstáculos en Big Data, ya que existe una gran cantidad de información difícil de entender y clasificar. Tomar en cuenta que no todos los datos son información y es necesario conocer si los datos a analizar tienen valor.
- Se puede mejorar el acceso del prototipo de usabilidad desarrollado para el manejo de las 4 herramientas de Hadoop, ya que su funcionamiento lo hace localmente sobre las herramientas instaladas en el ordenador, pero se puede mejorar su accesibilidad en medio de una red de computadores realizando una mayor investigación que permita el envío de comandos sobre las herramientas de Hadoop de manera remota.
- La instalación de Hadoop en un ordenador o servidor tiene un alto grado de complejidad. Es recomendable que el usuario se tome su tiempo para relacionarse con los temas, definiciones y arquitectura de Hadoop, con el fin de poder entender que es lo que se está realizando en cada paso de configuración del núcleo de procesamiento.
- En el caso de realizar la instalación y uso de nuevas herramientas de Hadoop, se recomienda analizar detenidamente requerimientos mínimos del sistema y alternativas de soporte técnico antes de implementar una nueva herramienta, debido a que no todas las herramientas funcionan de la misma manera en entornos similares.

BIBLIOGRAFÍA

- Agudelo, C. (2015). *BIG DATA ¿Qué es? Enormes cantidades de datos*. Retrieved September 10, 2016, from https://www.haikudeck.com/big-data-science-and-technology-presentation-oOMB2Oa4di#slide0
- Chen, C. (2015). *Diseño Apache Hive*. Retrieved October 22, 2016, from https://cwiki.apache.org/confluence/display/Hive/Design
- Cloudera, Inc. (2017). *Cloudera Enterprise Reference Architecture for AWS Deployments.* 1001 Page Mill Road, Building 2, 4-15.
- Deitel, P. & Deitel, H. (2013). *Como programar en Java*. México, Editorial ADDISON-WESLEY, 9a edición.
- Gartner (2012). Gartner IT Glossary: Big Data. Retrieved September 06, 2016, from http://www.gartner.com/it-glossary/big-data.
- Handytec. (2016). *Big Data y Data Analytics*. Retrieved September 07, 2016, from http://handytec.mobi/other/BrochureBigData.pdf
- Hurwitz, J., Nugent, A., Halper, F., & Kaufman, M. (2013). *Big Data for Dummies.* John Wiley & Sons, Inc., New Jersey.
- Isla, V. (2014). *Instalar Big Data paso a paso: Seguridad: Sentry, Ranger, Knox, Kerberos, LDAP & SSL*. España.
- Joyanes, L. (2013). Big Data. *Análisis De Grandes Volúmenes De Datos En Organizaciones*. Editorial AlfaOmega, Madrid.
- Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. USA, META Group Inc.
- Lublinsky, B. & Smith, Kevin (2014). *Hadoop. Soluciones Big Data*. Primera Edición, Editorial Anaya Multimedia/Wrox, Madrid.
- Manyika, J., Chul M., & Brown, M. (2011). *Big Data: The next frontier for innovation, competition and opportunity*. Mckinsey Global Intitute.
- Mayer-Schönberger, V. & Cukier, K. (2013). *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Nueva York, Editorial Houghton Mifflin Hartcourt, 2a edición.
- Nielsen, L. (2013). *Hadoop: The Engine That Drives Big Data (New Street Executive Summaries)*. Editorial New Street Communications, LLC (1709), USA.
- Pérez, J. & Asturiano, J. (2015). *Big Data: Oracle & Hadoop*. Retrieved January 15, 2017, from http://www.oracle.com/technetwork/es/articles/database-performance/big-data-oracle-hadoop-2813760-esa.html

- Piono, A. (2016). *Big Data y sus fuentes de datos: Guía rápida para elegir tu base de datos*. Editorial Kindle, Argentina.
- Salvador, F. (2014). *Big Data: ¿la ruta o el destino?*. Tecnología y crecimiento 3. IE Busines School, IE University. Departamento de Operaciones y Tecnología, Advance serie Foundation.
- Schnase, J., Duffy, D., Thompson, H., Nadeau, D., Sinno, S. y Strong, S. (2012). Applying Apache Hadoop to NASA's Big Climate Data, *Rev National Aeronautics and Space Administration*, 2-10.
- Schneider, R. (2014). *Hadoop for Dummies*. Compliments of IBM Platform Computing. USA.
- Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufano D. (2012). *Analytics: el uso de Big Data en el mundo real*. IBM Institute for Business Value, Escuela de Negocios Saïd en la Universidad de Oxford
- Slocum, M. (2011). *Big data now current perspectives de O'Reilly radar*. Sebastopol, CA: O'Reilly Media.
- Smolan, R. & Erwitt, J (2012). *The Human Face of Big Data*. Smolan & Ritter, Against All Odds Productions.
- The Apache Software Foundation. (2017). Apache Projects. Retrieved November 12, 2016, from http://www.apache.org/
- Universidad de Barcelona. (2015). *El impacto del Big-data en la Sociedad de la Información. Significado y utilidad*. España: Antonio Monleón-Getino. 428 – 440.
- Universidad de Mondragón. (2014). *Big Data y Hadoop. Cloudera vs Hortonworks*. España: Gorka Hurtado.
- White, T. (2015). Hadoop – The Definitive Guide. STORAGE AND ANALYSIS AT INTERNET SCALE. 4ta edición. O'Reilly.

ANEXOS

Anexo 1: Glosario de siglas y términos técnicos

- **Bit:** Unidad de medida de cantidad de información, equivalente a la elección entre dos posibilidades igualmente probables 0 - 1.
- **BPM:** Gestión de Procesos de Negocio (en inglés: Business Process Management o B.P.M.).
- **CDR:** extensión y formato nativo del programa CorelDRAW.
- **Clúster:** conjuntos o conglomerados de ordenadores unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.
- **CRM:** Gestión de relaciones con los clientes (en inglés: Customer Relationship Management).
- **EAR:** (en inglés: Enterprise Archive), son ficheros desplegables en servidores de aplicaciones que soporten el stack completo de JEE.
- **EJB:** (en inglés: Enterprise JavaBeans), son una de las interfaces de programación de aplicaciones (API) que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE).
- **ERP:** Sistemas de planificación de recursos empresariales (en inglés: Enterprise Resource Planning).
- **Georeferencial:** técnica de posicionamiento espacial de una entidad en una localización geográfica única y bien definida en un sistema de coordenadas y datum específicos.
- **GPS:** Sistema de Posicionamiento Global (en inglés: Global Positioning System).
- **JDBC:** Java Database Connectivity es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo.
- **JSON:** JavaScript Object Notation es un formato para el intercambios de datos, que los describe con una sintaxis dedicada que se usa para identificar y gestionar los datos.
- **Localhost:** hace referencia a un ambiente de red de la computadora local donde la aplicación está corriendo.
- **Metadatos:** datos que describen otros datos con el fin de ayudar a identificar objetos que contienen información valiosa dentro de un contexto determinado.
- **MySQL:** sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation.
- **Nodo:** punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar.

- **ODBC:** estándar de acceso a bases de datos, que permite mantener independencia entre los lenguajes de programación.
- **Open Source:** también llamado "Código Abierto", término que se utiliza para denominar a cierto tipo de software que se distribuye mediante una licencia que le permite al usuario final utilizar el código fuente del programa para estudiarlo, modificarlo y realizar mejoras en el mismo.
- **Querys:** concepto en el idioma inglés traducido al español como consultas realizadas contra una base de datos.
- **Runtime:** es un conjunto de utilidades que permite la ejecución de programas Java.
- **SMS:** servicio de mensajes cortos o servicio de mensajes simples (en inglés: Short Message Service).
- **Streaming:** es la distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se descarga.
- **Software:** conjunto de programas e instrucciones que permiten llevar a cabo diferentes tareas dentro de una computadora.
- **Twitter:** red social en línea que permite a los usuarios enviar y leer mensajes cortos de 140 caracteres llamados "tweets".
- **WAR:** (en inglés: Web Application Archive), es un JAR para manejo de aplicaciones WEB que contienen .class, JSP's, Servlets, entre otros.
- **XML:** (en inglés: Extensible Markup Language), es un lenguaje que permite la organización y el etiquetado de documentos.

Anexo 2: Instalación de Java

Se debe seguir los siguientes pasos para la instalación de Java en Linux:

1. Actualizar el sistema operativo Linux Ubuntu con las últimas actualizaciones del sistema utilizando el siguiente comando en la terminal (ingresar el password de súper usuario cuando lo solicite):

```
$ sudo apt-get update
```

Figura 208: Instalación Java Paso 1
Fuente: Elaboración propia.

2. Instalar la versión de Java que se descargó en las actualizaciones realizadas en el paso 1. Se ejecutará el siguiente comando en la terminal:

```
$ sudo apt-get install default-jdk
```

Figura 209: Instalación Java Paso 2
Fuente: Elaboración propia.

Si existen preguntas al momento de la instalación se debe ingresar la letra Y (yes).

3. Al realizar los pasos 1 y 2 se instalará exitosamente la versión de Java. Se lo confirma haciendo uso del siguiente comando:

```
$ java -version
```

Figura 210: Instalación Java Paso 3.1
Fuente: Elaboración propia.

Al ejecutar el comando en la terminal de la máquina virtual se obtuvo el siguiente resultado:

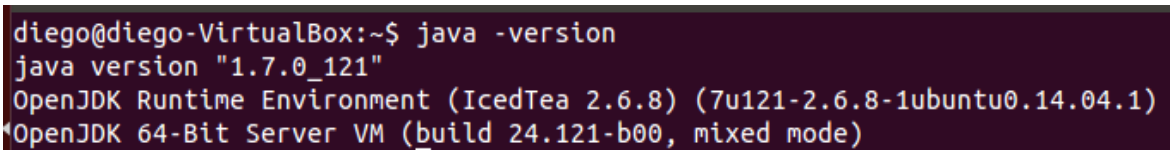


Figura 211: Instalación Java Paso 3.2
Fuente: Elaboración propia.

Anexo 3: Instalación de Hadoop

Para la instalación de Hadoop en Linux se hace uso de la versión de Hadoop 2.6.0, por lo cual se debe seguir los siguientes pasos:

Creación de usuario de Hadoop:

Previo a la instalación de Hadoop, es necesario crear un usuario con permisos de administrador que sea de uso exclusivo de Hadoop, ya que de esta manera se busca el aislar el sistema de archivos de Hadoop con el de sistema de archivos de Unix.

Se creará un grupo llamado Hadoop en el que se agregará el usuario hduser (nuevo usuario para Hadoop). Se debe ejecutar los siguientes comandos:

```
$ sudo addgroup hadoop
```

Figura 212: Creación de grupo de Hadoop
Fuente: Elaboración propia.

```
$ sudo adduser --ingroup hadoop hduser
```

Figura 213: Creación de usuario de Hadoop
Fuente: Elaboración propia.

Al crear el usuario hduser se solicitará obligatoriamente el nuevo password e información adicional (nombres, apellidos, teléfono) la misma que es opcional ingresarla. Si existen preguntas al momento de la creación del usuario se debe ingresar la letra Y (yes).

Configuración SSH y la generación de claves

Se debe realizar una configuración SSH (Secure SHell, en español: intérprete de órdenes seguro) la misma que se utilizará para realizar varias operaciones sobre un clúster tales como: iniciar, detener y operar el shell en forma distribuida. En nuestro caso, es necesario autenticar el nuevo usuario de Hadoop (hduser) proporcionándole un par de claves público/privado que permita compartir con diferentes usuarios.

Para instalar el SSH se debe ejecutar el siguiente comando en la terminal del sistema:

```
$ sudo apt-get install ssh
```

Figura 214: Instalación de SSH
Fuente: Elaboración propia.

Si existen preguntas al momento de la instalación de SSH se debe ingresar la letra Y (yes).

A continuación, se va a otorgar permisos de súper administrador al usuario de Hadoop (hduser) utilizando el siguiente comando en la terminal del sistema:

```
$ sudo adduser hduser sudo
```

Figura 215: Permisos súper administrador a usuario Hadoop
Fuente: Elaboración propia.

Para finalizar, se deberá generar las claves pública/privada para el usuario de Hadoop (hduser). Por lo cual, se deberá ejecutar los siguientes comandos utilizando el usuario de Hadoop en el siguiente orden:

```
$ su hduser
password: (contraseña usuario)
```

Figura 216: Ingreso al sistema con usuario hduser
Fuente: Elaboración propia.

```
$ ssh-keygen -t rsa -P ""
```

Figura 217: Generación de clave SSH
Fuente: Elaboración propia.

```
$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Figura 218: Permisos para clave SSH
Fuente: Elaboración propia.

```
$ ssh localhost
```

Figura 219: Agregar SSH en localhost
Fuente: Elaboración propia.

Si existen preguntas al momento de agregar el SSH al localhost se debe ingresar la palabra yes.

Descarga de Hadoop:

La descarga y descompresión de los datos descargados se lo realizará ejecutando los siguientes comandos en la terminal:

```
$ su hduser
password: (contraseña usuario)
$ wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
$ tar xvzf hadoop-2.6.0.tar.gz
$ cd hadoop-2.6.0
$ sudo mkdir /usr/local/hadoop
$ sudo mv * /usr/local/hadoop
```

Figura 220: Descarga y descompresión de Hadoop
Fuente: Elaboración propia.

Luego de la descarga de Hadoop, se debe tomar en cuenta los modos de funcionamiento de Hadoop previo a su instalación:

- **Modo Autónomo Local:** Posterior a la descarga de Hadoop en el sistema Linux, de forma predeterminada Hadoop se configura en modo autónomo el cual se ejecutará en un solo proceso java.
- **Pseudo Modo Distribuido:** Es una configuración de Hadoop que realiza una simulación distribuida; es decir cada proceso de Hadoop (HDFS, Yarn, MapReduce, etc.) se ejecutará en un proceso java independiente. Este modo es utilizado frecuentemente para realizar desarrollos.
- **Modo Totalmente Distribuido:** Este modo de funcionamiento es completamente distribuido y se basa en el funcionamiento de un clúster en el que es necesario la utilización de dos o más máquinas.

Nota: Por motivos de estudio se ha considerado la utilización del modo de funcionamiento de Hadoop 2.6.0 en Pseudo Modo Distribuido, ya que el mismo se ajusta a las funcionalidades que se van a tratar para cumplir con los objetivos trazados en el presente trabajo de titulación.

Instalación de Hadoop en Pseudo Modo Distribuido:

Para la instalación de Hadoop 2.6.0 en Pseudo Modo Distribuido debe modificarse 5 archivos de configuración propios del sistema como se describe a continuación:

1. Se debe editar el archivo bashrc del sistema operativo, utilizando el siguiente comando en la terminal:

```
$ vim ~/.bashrc
```

Figura 221: Instalación Hadoop Pseudo Modo Distribuido Paso 1.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Hadoop 2.6.0. Las variables a agregar son:

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

Figura 222: Instalación Hadoop Pseudo Modo Distribuido Paso 1.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo bashrc de la máquina virtual quedó configurada de la siguiente manera:

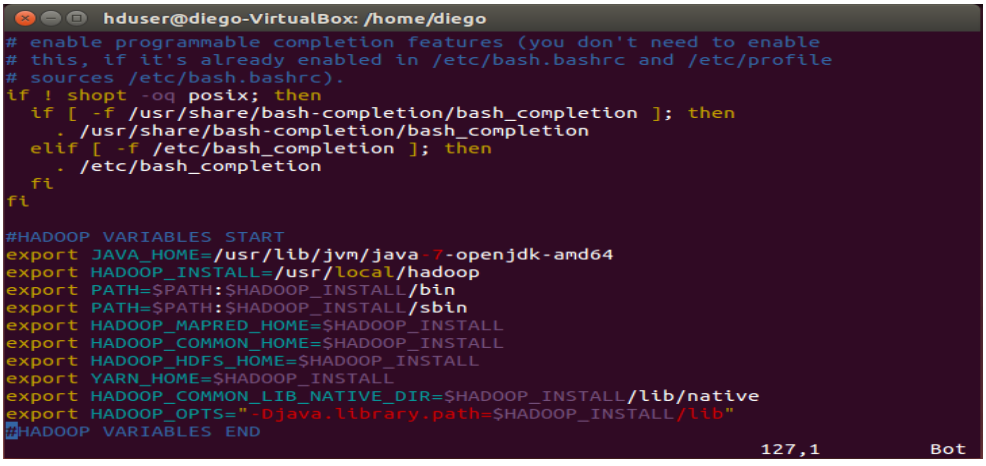


Figura 223: Instalación Hadoop Pseudo Modo Distribuido Paso 1.3
Fuente: Elaboración propia.

2. El siguiente archivo a editar es el encargado del ambiente de Hadoop y tiene como nombre hadoop-env.sh. Su edición dependerá del directorio donde fue descomprimido Hadoop. Se debe ejecutar el siguiente comando en la terminal:

```
$ vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Figura 224: Instalación Hadoop Pseudo Modo Distribuido Paso 2.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las variables de configuración de Java:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

Figura 225: Instalación Hadoop Pseudo Modo Distribuido Paso 2.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo hadoop-env.sh de la máquina virtual quedó configurada de la siguiente manera:

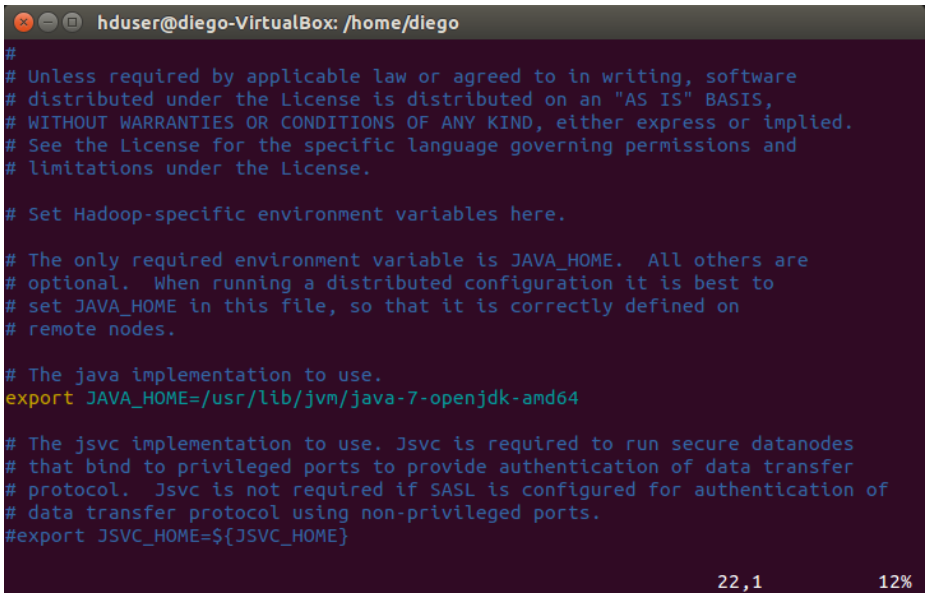


Figura 226: Instalación Hadoop Pseudo Modo Distribuido Paso 2.3
Fuente: Elaboración propia.

3. El siguiente archivo a editar tiene como nombrecore-site.xml, el cual contiene la información del número del puerto usado por la instancia de Hadoop en la que se muestra la memoria asignada para el sistema de archivos, el límite de memoria que se tiene asignada para almacenar los datos y el tamaño de lectura/escritura.

Para su edición se debe crear un directorio temporal y proporcionar permisos para manipular el archivo sobre este directorio temporal. Se debe ejecutar los siguientes comandos en la terminal:

```
$ sudo mkdir -p /app/hadoop/tmp
$ sudo chown hduser:hadoop /app/hadoop/tmp
$ vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

Figura 227: Instalación Hadoop Pseudo Modo Distribuido Paso 3.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las siguientes propiedades:

```
<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://localhost:54310</value>
 <description>The name of the default file system.  A URI whose
 scheme and authority determine the FileSystem implementation.  The
 uri's scheme determines the config property (fs.SCHEME.impl) naming
 the FileSystem implementation class.  The uri's authority is used to
 determine the host, port, etc. for a filesystem.</description>
</property>
```

Figura 228: Instalación Hadoop Pseudo Modo Distribuido Paso 3.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo core-site.xml de la máquina virtual quedó configurada de la siguiente manera:

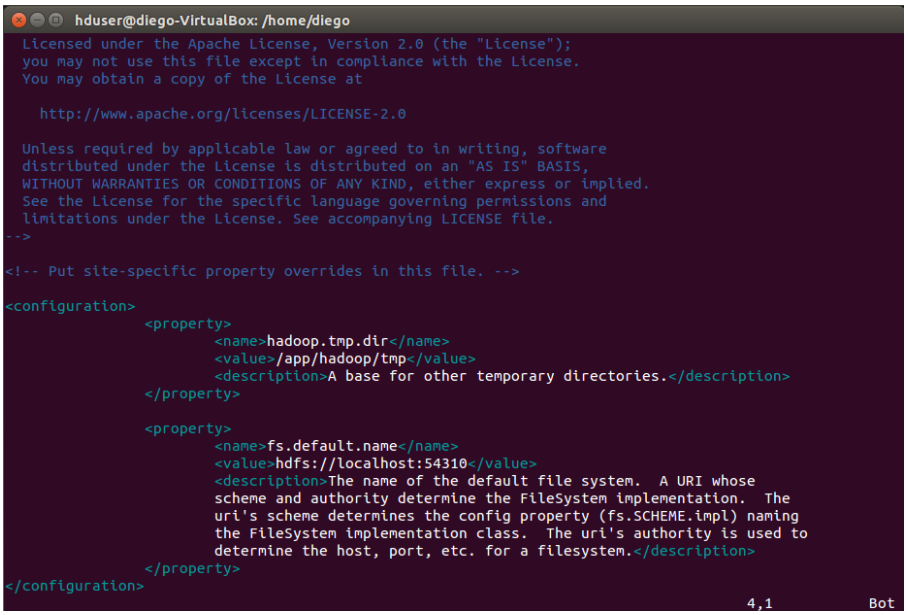


Figura 229: Instalación Hadoop Pseudo Modo Distribuido Paso 3.3
Fuente: Elaboración propia.

4. El siguiente archivo a editar tiene como nombre mapred-site.xml. En este archivo se especifica el framework del MapReduce que se utilizará. Para poder configurar el mapred-site.xml se debe copiar el archivo mapred-site, xml.template dependiendo del directorio donde fue descomprimido Hadoop y posteriormente debe ser editado. Se debe ejecutar los siguientes comandos en la terminal:

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

Figura 230: Instalación Hadoop Pseudo Modo Distribuido Paso 4.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las siguientes propiedades:

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs at. If "local", then jobs are run in-process as a single map and reduce task. </description>
</property>
```

Figura 231: Instalación Hadoop Pseudo Modo Distribuido Paso 4.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo mapred-site.xml de la máquina virtual quedó configurada de la siguiente manera:

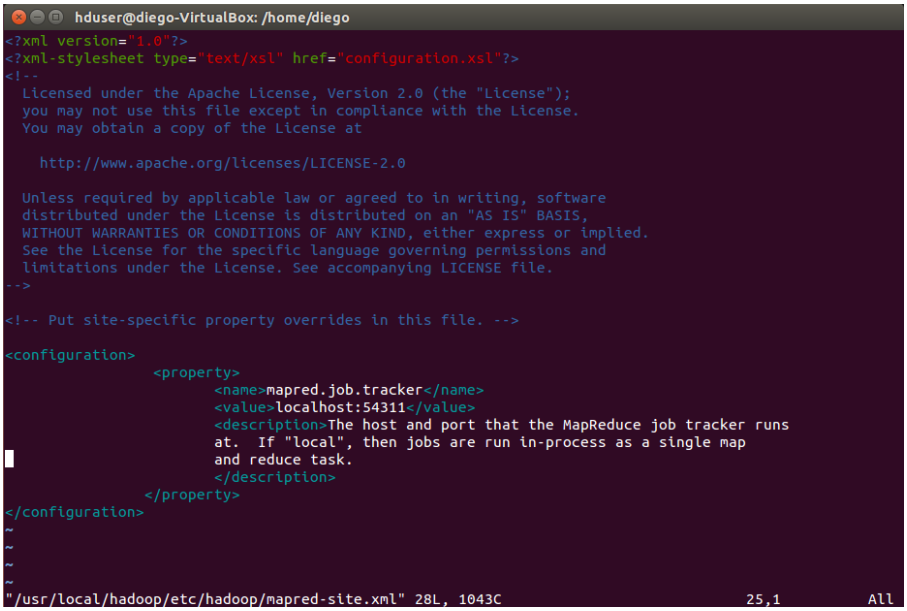


Figura 232: Instalación Hadoop Pseudo Modo Distribuido Paso 4.3
Fuente: Elaboración propia.

5. El siguiente archivo a editar tiene como nombre hdfs-site.xml, el cual posee la información del valor de los datos de réplica, la ubicación del nodo y las rutas de accesos de los sistemas de archivos locales de los nodos.

Para poder configurar el hdfs-site.xml se debe ejecutar los siguientes comandos en la terminal:

```
$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
$ sudo chown -R hduser:hadoop /usr/local/hadoop_store
$ vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Figura 233: Instalación Hadoop Pseudo Modo Distribuido Paso 5.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las siguientes propiedades:

```
<property>
 <name>dfs.replication</name>
 <value>1</value>
 <description>Default block replication.
 The actual number of replications can be specified when the file is created.
 The default is used if replication is not specified in create time.
 </description>
</property>

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

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

Figura 234: Instalación Hadoop Pseudo Modo Distribuido Paso 5.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo hdfs-site.xml de la máquina virtual quedó configurada de la siguiente manera:

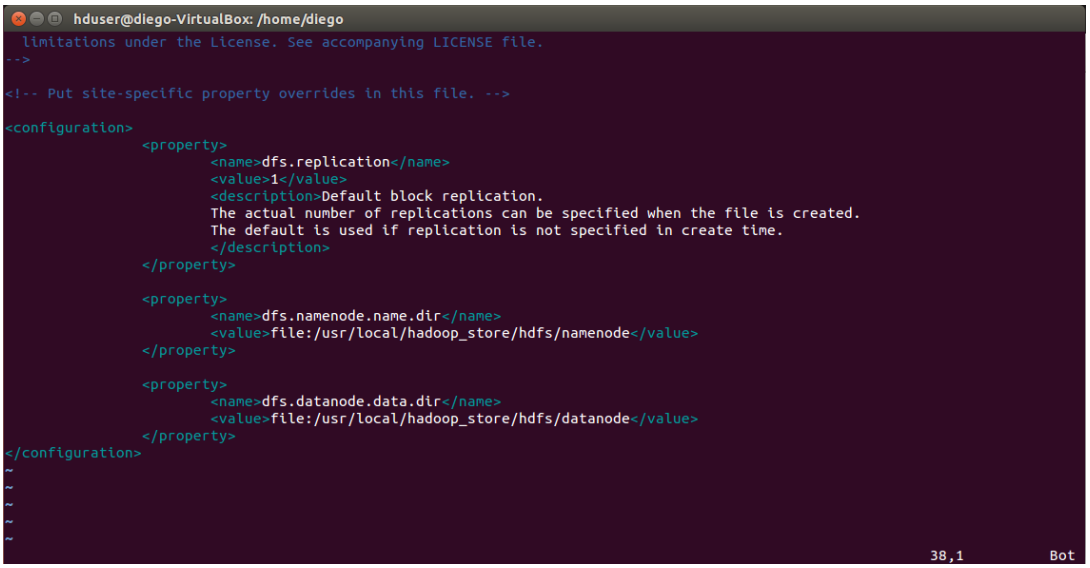


Figura 235: Instalación Hadoop Pseudo Modo Distribuido Paso 5.3
Fuente: Elaboración propia.

6. Para que se apliquen los cambios realizados sobre los5 archivos de configuración propios de Hadoop se debe reiniciar el sistema operativo Linux Ubuntu de la máquina virtual.

7. Al reiniciarse el sistema operativo Linux Ubuntu, se debe que ejecutar la terminal e ingresar el usuario hduser con su password. A continuación, se debe darle formato al NameNode de Hadoop, con lo cual se confirmaría que se instaló correctamente Hadoop 2.6.0 en Linux Ubuntu. Para poder terminar de configurar Hadoop se debe ejecutar los siguientes comandos en la terminal:

```
$ su hduser
password: (contraseña usuario)
$ hadoop namenode -format
```

Figura 236: Instalación Hadoop Pseudo Modo Distribuido Paso 7.1
Fuente: Elaboración propia.

Si se configuró correctamente Hadoop 2.6.0, en la pantalla de la terminal se mostrará la creación y configuración de los nodos que se utilizarán para su funcionamiento.
Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

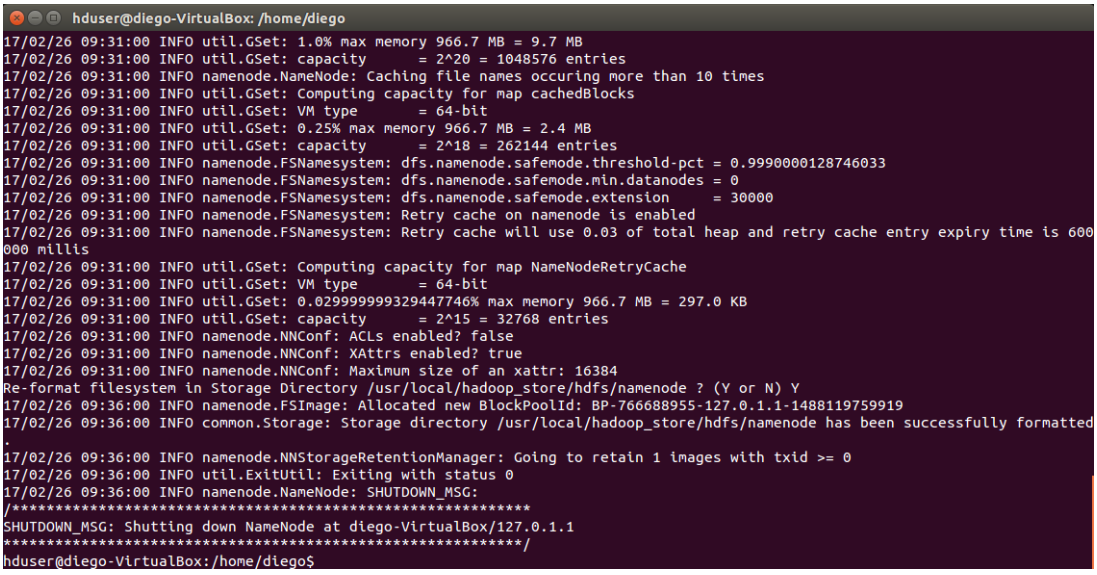


Figura 237: Instalación Hadoop Pseudo Modo Distribuido Paso 7.2
Fuente: Elaboración propia.

Verificación de la instalación de Hadoop:

Se debe ejecutar los siguientes pasos para confirmar que se instaló correctamente Hadoop en el sistema Linux:

8. Para poder iniciar Hadoop 2.6.0 en Linux, se debe estar registrado en la terminal con el usuario hduser y se debe darle permisos de cambio de propietario en archivos de Hadoop. Para poder dar los permisos aplicar el siguiente comando en la terminal:

```
$ sudo chown -R hduser:hadoop /usr/local/hadoop/
```

Figura 238: Verificación Hadoop Paso 8
Fuente: Elaboración propia.

9. Dirigirse a la ubicación de la carpeta sbin de Hadoop, en donde se lista todos los archivos que contiene la carpeta. Uno de los archivos que debe contener es el llamado start-all.sh, con lo cual se confirmaría que se está en la ubicación correcta. Se debe ejecutar el siguiente comando sobre la terminal.

```
$ cd /usr/local/hadoop/sbin
$ ls
```

Figura 239: Verificación Hadoop Paso 9.1
Fuente: Elaboración propia.

Al ejecutar los comandos en la máquina virtual se mostró de la siguiente manera:

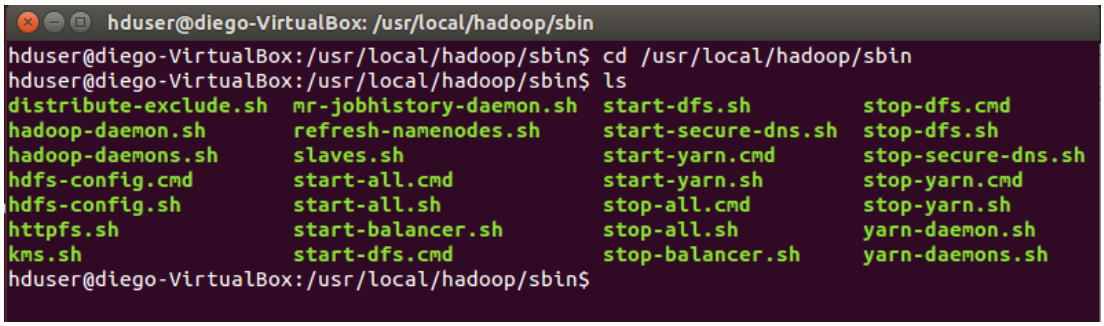


Figura 240: Verificación Hadoop Paso 9.2
Fuente: Elaboración propia.

10. Iniciar los servicios de Hadoop. Para poder hacerlo se debe ejecutar en la terminal del sistema el comando start-all.sh:

```
$ start-all.sh
```

Figura 241: Verificación Hadoop Paso 10.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

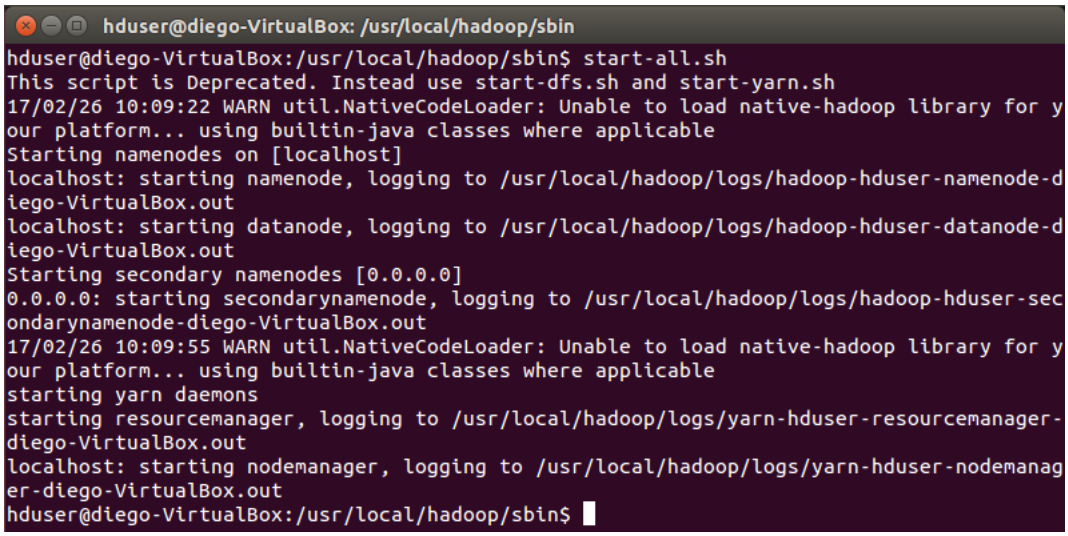


Figura 242: Verificación Hadoop Paso 10.2
Fuente: Elaboración propia.

Se confirma que los servicios se levantaron correctamente ejecutando el siguiente comando en la terminal:

```
$ jps
```

Figura 243: Verificación Hadoop Paso 10.3
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

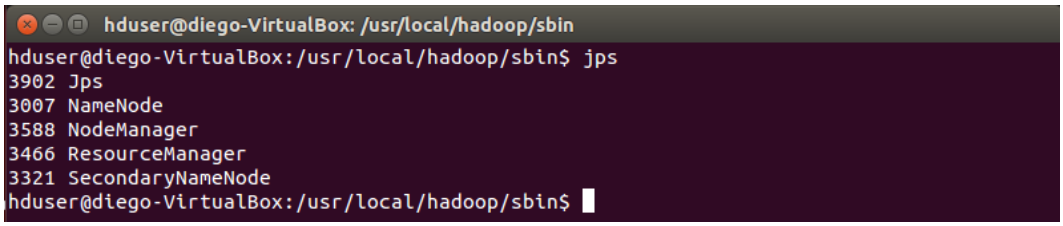


Figura 244: Verificación Hadoop Paso 10.4
Fuente: Elaboración propia.

11. A continuación, se debe acceder a Hadoop desde el navegador. El puerto predeterminado para acceder a Hadoop es el 50070 y se debe utilizar la siguiente url http://localhost:50070/ para obtener los servicios de Hadoop en el navegador.

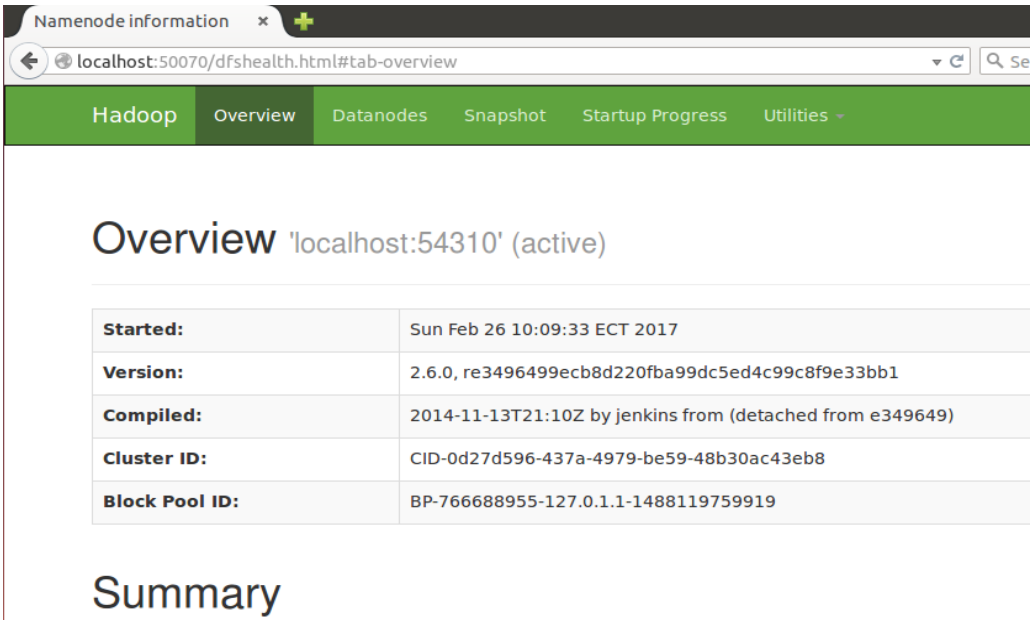


Figura 245: Verificación Hadoop Paso 11
Fuente: Elaboración propia.

12. Como paso final, se debe verificar todas las aplicaciones del clúster. El puerto predeterminado para acceder a todas las aplicaciones del clúster es el 8088 y se debe utilizar la siguiente url http://localhost:8088/ para poder visualizar este servicio.

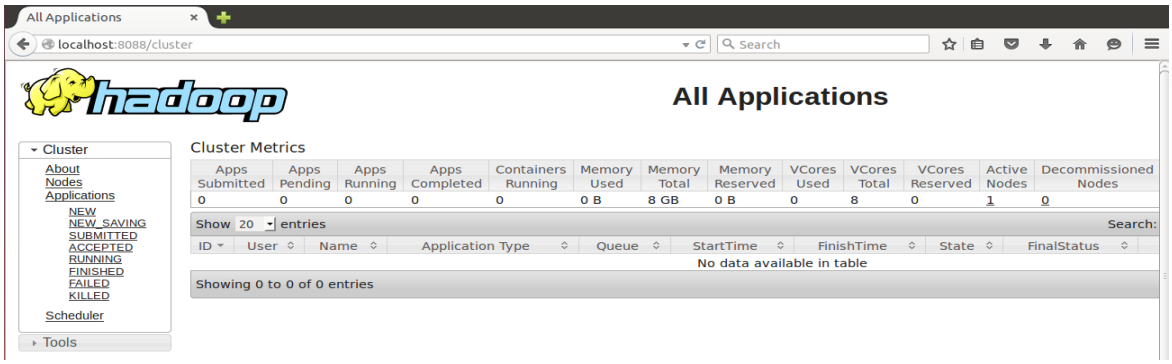


Figura 246: Verificación Hadoop Paso 12
Fuente: Elaboración propia.

13. Luego de iniciar Hadoop, se detiene su servicio utilizando el siguiente comando en la terminal:

```
$ stop-all.sh
```

Figura 247: Verificación Hadoop Paso 13.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

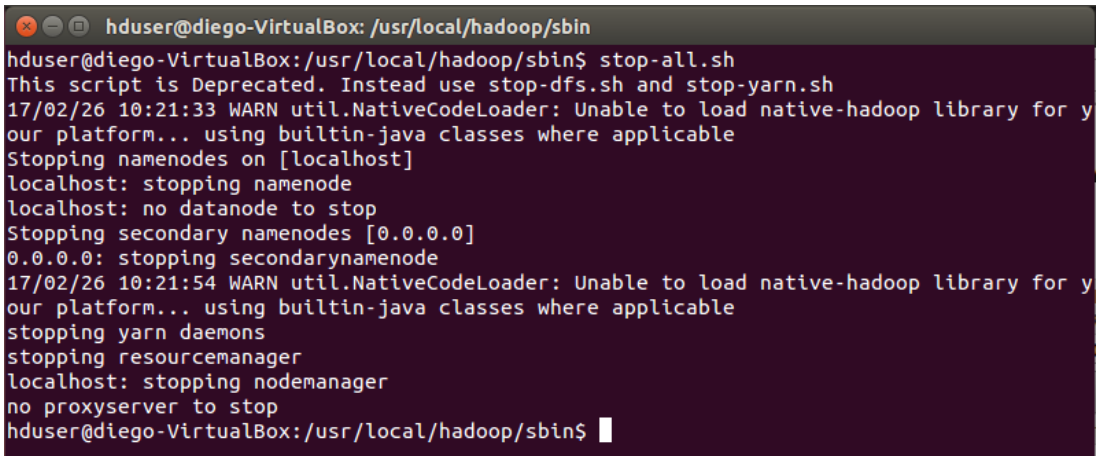


Figura 248: Verificación Hadoop Paso 13.2
Fuente: Elaboración propia.

Anexo 4: Instalación de Flume

La versión de Flume a instalar es la 1.6.0 y es necesario validar el sistema operativo en el que se va a ejecutar. Para Flume se recomienda el sistema operativo de Linux Ubuntu versión 14.04 LTS**,** ya que se lo puede utilizar para el desarrollo y despliegue de aplicaciones.

La instalación de Flume 1.6.0 según (White, 2015) se lo realiza de la siguiente manera:

Comprobación de la instalación de Java:

1. Se debe verificar si se tiene instalado en entorno Java en el sistema operativo haciendo uso del siguiente comando en la terminal:

```
$ java -version
```

Figura 249: Instalación Flume Paso 1.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

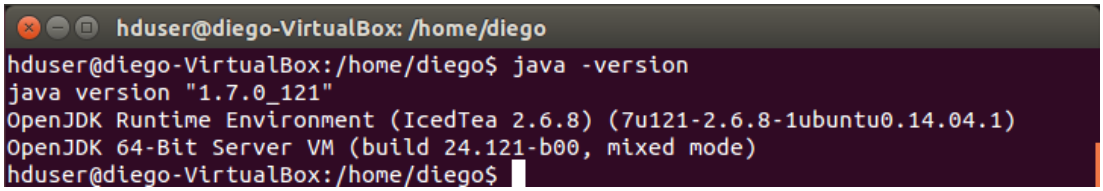


Figura 250: Instalación Flume Paso 1.2
Fuente: Elaboración propia.

En el caso de no tener instalado el entorno de Java seguir los pasos que se indican en el Anexo 2.

Comprobación de la instalación e inicio de Hadoop:

2. Se debe verificar la instalación Hadoop en el sistema haciendo uso del siguiente comando en la terminal del sistema:

```
$ hadoop version
```

Figura 251: Instalación Flume Paso 2.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

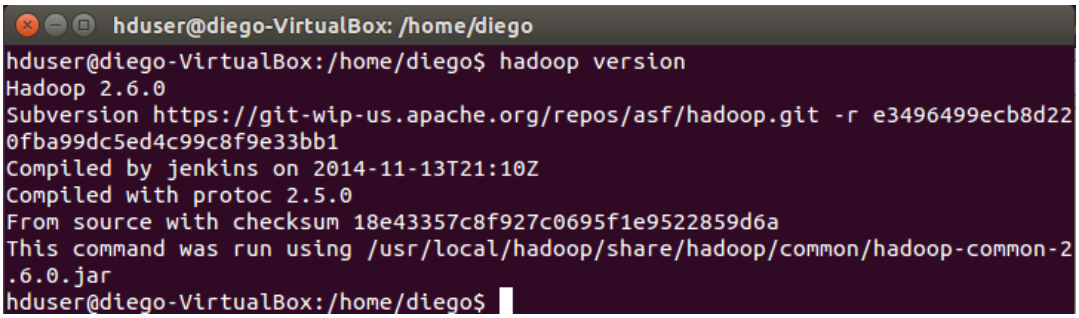


Figura 252: Instalación Flume Paso 2.2
Fuente: Elaboración propia.

Si está instalado correctamente Hadoop se debe iniciar el servicio como los puntos del subtema **Verificación de la instalación de Hadoop** en el Anexo 3, caso contrario seguir todos los pasos que conforman la instalación e inicio de Hadoop en el Anexo 3.

Instalación de Flume:

3. Se debe crear una carpeta con el nombre "work" en el directorio Home, haciendo uso del siguiente comando en la terminal:

```
$ mkdir work
```

Figura 253: Instalación Flume Paso 3.1
Fuente: Elaboración propia

Al ejecutar el comando en la máquina virtual se observa que la carpeta se ha creado de la siguiente manera:

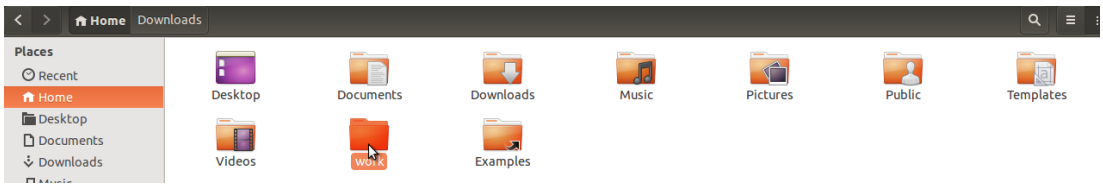


Figura 254: Instalación Flume Paso 3.2
Fuente: Elaboración propia

4. Se procede a descargar la versión de Flume 1.6.0 en la carpeta "work" creada en el paso 3, para lo cual ingresaremos al link http://archive.apache.org/dist/flume/1.6.0/ y se debe seleccionar apache-flume-1.6.0-bin.tar.gz. Como referencia se muestra la siguiente pantalla:

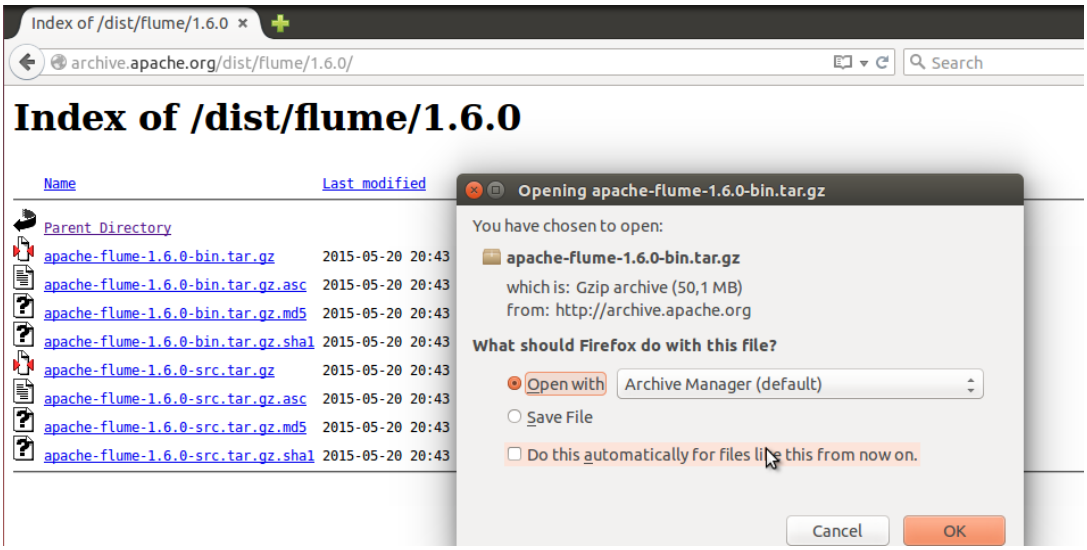


Figura 255: Instalación Flume Paso 4.1
Fuente: Elaboración propia

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "work" se ha descargado el archivo apache-flume-1.6.0-bin.tar.gz como se lo indica de la siguiente manera:

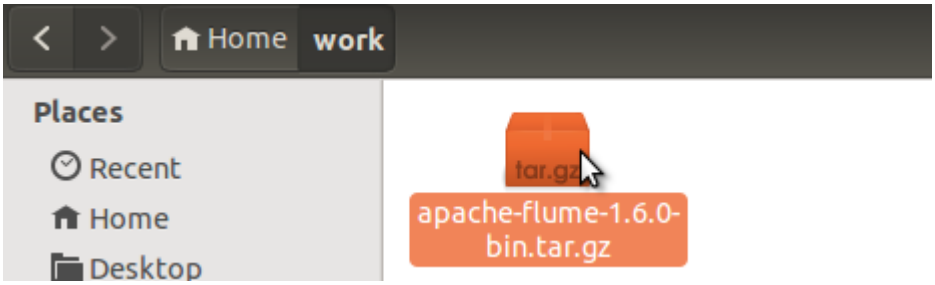


Figura 256: Instalación Flume Paso 4.2
Fuente: Elaboración propia

5. En la carpeta work se procede a extraer el archivo .tar de Flume 1.6.0 utilizando el siguiente comando en la terminal:

```
$ tar -xvzf apache-flume-1.6.0-bin.tar.gz
```

Figura 257: Instalación Flume Paso 5.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "work" se ha descomprimido el archivo y se ha creado la carpeta apache-flume-1.6.0-bin como se lo indica a continuación:

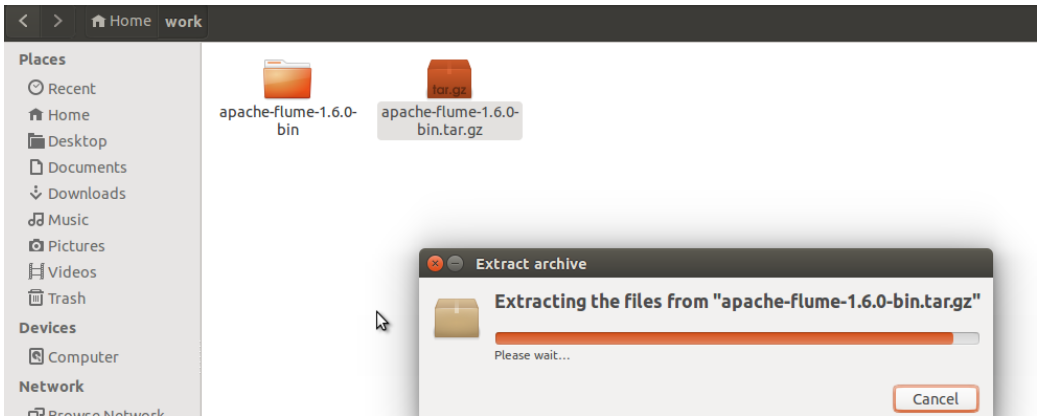


Figura 258: Instalación Flume Paso 5.2
Fuente: Elaboración propia.

6. Descargar la librería flume-sources-1.0-SNAPSHOT.jar ejecutando el siguiente comando en la terminal:

```
$ wget http://files.cloudera.com/samples/flume-source
s-1.0-SNAPSHOT.jar
```

Figura 259: Instalación Flume Paso 6.1
Fuente: Elaboración propia.

Al descargar el archivo, debe ser movido al directorio apache-flume-1.6.0-bin/lib. En la máquina virtual se visualiza el archivo en el directorio especificado:

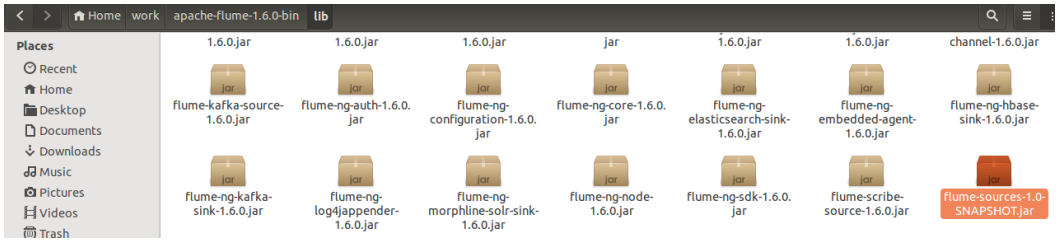


Figura 260: Instalación Flume Paso 6.2
Fuente: Elaboración propia.

Configuración de Flume:

Para configurar Flume en nuestro sistema se debe modificar los siguientes 2 archivos: flume-env.sh y bashrc.

7. En el archivo .bashrc se debe configurar la carpeta de inicio, y las rutas de clase para Flume utilizando el siguiente comando en la terminal:

```
$ vim ~/.bashrc
```

Figura 261: Configuración de Flume Paso 7.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Flume 1.6.0. Las variables a agregar son:

```
export FLUME_HOME=/home/hadoop/work/apache-flume-1.6.0-bin
export FLUME_CONF_DIR=$FLUME_HOME/conf
export FLUME_CLASS_PATH=$FLUME_CONF_DIR
export PATH=$FLUME_HOME/bin:$PATH
```

Figura 262: Configuración de Flume Paso 7.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo bashrc de la máquina virtual quedó de la siguiente manera:

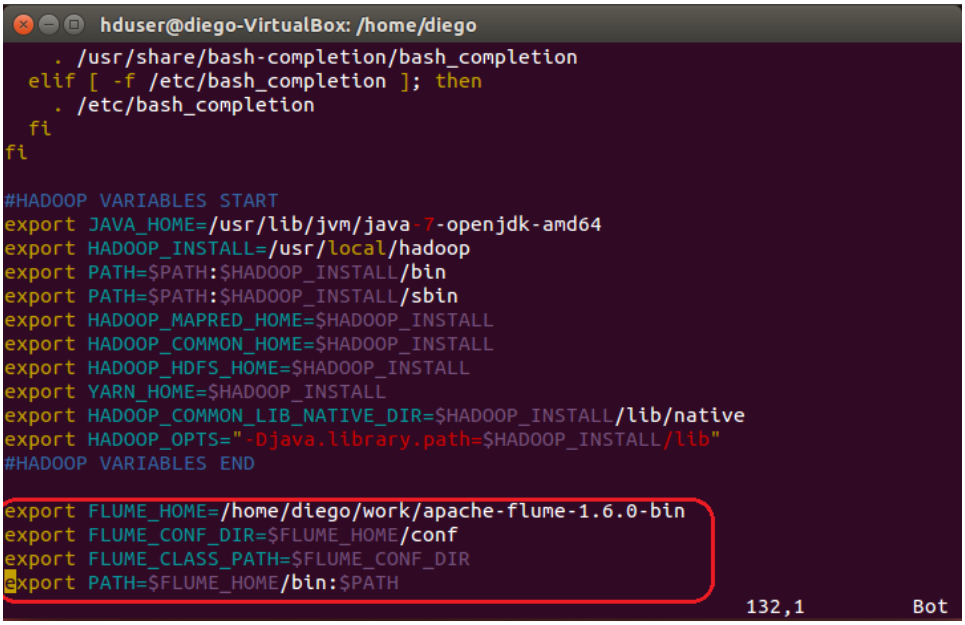


Figura 263: Configuración de Flume Paso 7.3
Fuente: Elaboración propia.

8. En el directorio Home/work/apache-flume-1.6.0-bin/conf existen los siguientes cuatro archivos: flume-conf.properties.template, flume-env.sh.template, flume-env.ps1.template, y log4j.properties. Tal como se muestra a continuación:

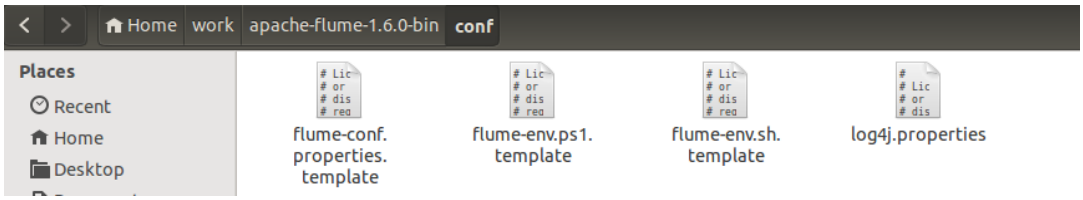


Figura 264: Instalación Flume Paso 8.1
Fuente: Elaboración propia.

De estos archivos, se debe crear una copia de flume-env.sh.template, al cual se le debe asignar el nombre de flume-env.sh. Como referencia se muestra en la imagen:

Figura 265: Instalación Flume Paso 8.2
Fuente: Elaboración propia.

Se debe editar el archivo flume-env.sh, al cual se le debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Flume 1.6.0. Las variables a agregar son:

```
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64/
FLUME_CLASSPATH="/home/hadoop/work/apache-flume-1.6.0-bin/lib/flume-sources-1.0-SNAPSHOT.jar"
```

Figura 266: Instalación Flume Paso 8.3
Fuente: Elaboración propia.

La configuración aplicada en el archivo flume-env.sh de la máquina virtual quedó de la siguiente manera:

```
# Enviroment variables can be set here.

# export JAVA_HOME=/usr/lib/jvm/java-6-sun

# Give Flume more memory and pre-allocate, enable remote monitoring via JMX
# export JAVA_OPTS="-Xms100m -Xmx2000m -Dcom.sun.management.jmxremote"

# Note that the Flume conf directory is always included in the classpath.
#FLUME_CLASSPATH=""
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64/
FLUME_CLASSPATH="/home/diego/work/apache-flume-1.6.0-bin/lib/flume-sources-1.0-SNAPSHOT.jar"
```

Figura 267: Instalación Flume Paso 8.4
Fuente: Elaboración propia.

Verificación de Flume:

9. Para confirmar que Flume se encuentra instalado correctamente en el sistema se debe ingresar al directorio Home/work/apache-flume-1.6.0-bin/bin y ejecutar el siguiente comando en la terminal del sistema:

```
$ ./flume-ng
```

Figura 268: Instalación Flume Paso 9.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se debe tener un resultado como el que se muestra a continuación:

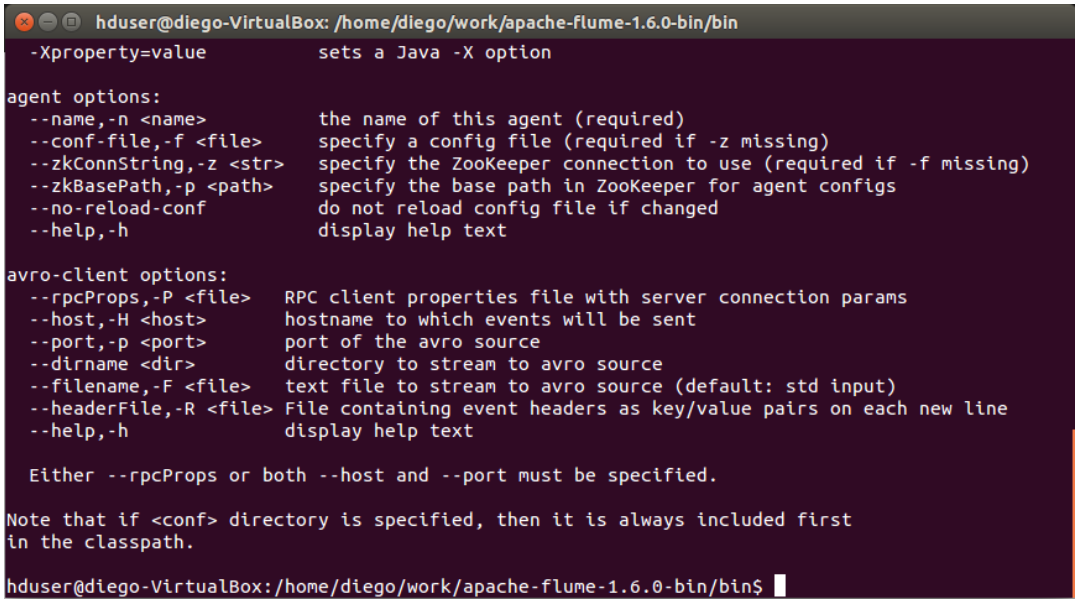


Figura 269: Instalación Flume Paso 9.2
Fuente: Elaboración propia.

Anexo 5: Instalación de Hive

Previo a la instalación de Hive es necesario validar el sistema operativo en el que se va a ejecutar. Para Hive se recomienda cualquier sistema operativo de Linux**,** ya que se lo utilizará para el desarrollo y despliegue de aplicaciones.

La instalación de Hive según (White, 2015) se lo realiza de la siguiente manera:

Comprobación de la instalación de Java:

1. Se debe verificar si se tiene instalado en entorno Java en el sistema haciendo uso del siguiente comando en la terminal:

```
$ java -version
```

Figura 270: Instalación Hive Paso 1.1
Fuente: Elaboración propia.

Con el comando anterior se observa que la versión de Java instalada en el sistema como se muestra a continuación, caso contrario seguir los pasos que se indican en el Anexo 2.

```
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b13)
Java HotSpot(TM) Client VM (build 25.0-b02, mixed mode)
```

Figura 271: Instalación Hive Paso 1.2
Fuente: Elaboración propia.

Comprobación de la instalación de Hadoop en Pseudo Modo Distribuido:

2. Se debe verificar si se tiene instalado Hadoop en el sistema haciendo uso del siguiente comando en la terminal del sistema:

```
$ hadoop version
```

Figura 272: Instalación Hive Paso 2.1
Fuente: Elaboración propia.

Con el comando anterior se observa en la siguiente figura la versión de Hadoop instalada, caso contrario seguir los pasos que se indican en el Anexo 3 para la instalación de Hadoop en Pseudo Modo Distribuido.

```
Hadoop 2.4.1 Subversion https://svn.apache.org/repos/asf/hadoop/common -r 1529768
Compiled by hortonmu on 2013-10-07T06:28Z
Compiled with protoc 2.5.0
From source with checksum 79e53ce7994d1628b240f09af91e1af4
```

Figura 273: Instalación Hive Paso 2.2
Fuente: Elaboración propia.

Descarga de Hive:

3. Se debe descargar la última versión de Hive, para lo cual ingresaremos al link http://www-eu.apache.org/dist/hive/ (al momento la última versión de Hive es la 2.1.0 y se deben descargar apache-hive-2.1.0-bin.tar.gz).

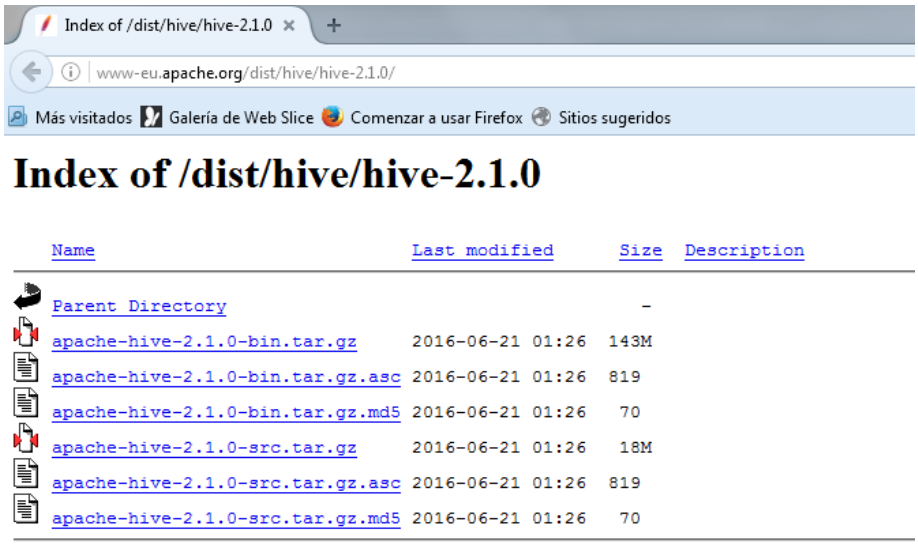


Figura 274: Instalación Hive Paso 3
Fuente: The Apache Software Foundation. (2017).

Instalación de Hive:

4. Extraer el archivo tar de Hive descargado en la carpeta de nombre "Downloads" utilizando el siguiente comando en la terminal:

```
$ cd Downloads
$ tar zxvf apache-hive-2.0.1-bin.tar.gz
```

Figura 275: Instalación Hive Paso 4
Fuente: Elaboración propia.

5. Se debe copiar los archivos extraídos en el paso 4, al directorio /usr/local/Hive utilizando el siguiente comando en la terminal:

```
$ su -
passwd: (contraseña root)

# cd /home/user/Download
# mv apache-hive-2.0.1-bin /usr/local/hive
# exit
```

Figura 276: Instalación Hive Paso 5
Fuente: Elaboración propia.

6. Configurar el medio ambiente de Hive adicionando las siguientes líneas en el archivo ~/.bashrc:

```
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/*:.
export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/*:.
```

Figura 277: Instalación Hive Paso 6
Fuente: Elaboración propia.

7. A continuación, se debe aplicar los cambios en el sistema:

```
$ source ~/.bashrc
```

Figura 278: Instalación Hive Paso 7
Fuente: Elaboración propia.

Configuración de Hive con Hadoop:

8. Es necesario configurar Hive con Hadoop. Se debe editar el archivo hive-env.sh que se ubica en el directorio $HIVE_HOME/conf. Se debe ejecutar el siguiente en la terminal para que se realice la redirección y copia del archivo de plantilla de Hive:

```
$ cd $HIVE_HOME/conf
$ cp hive-env.sh.template hive-env.sh
```

Figura 279: Instalación Hive Paso 8.1
Fuente: Elaboración propia.

A continuación, se debe editar el archivo hive-env.sh agregando la siguiente línea:

```
export HADOOP_HOME=/usr/local/hadoop
```

Figura 280: Instalación Hive Paso 8.2
Fuente: Elaboración propia.

Con este último paso se ha completado exitosamente la instalación de Hive. A continuación se va a configurar un servidor de base de datos externo para configurar el Metastore.

Descarga e Instalación de Apache Derby:

Se debe seguir los siguientes pasos para descargar e instalar Apache Derby dentro de Hive:

9. Se debe descargar Apache Derby ejecutando el siguiente comando en la terminal:

```
$ cd Downloads
$ wget http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin.tar.gz
```

Figura 281: Instalación Hive Paso 9
Fuente: Elaboración propia.

10. Extraer el archivo tar de Derby descargado en la carpeta Downloads utilizando el siguiente comando en la terminal:

```
$ cd Downloads
$ tar zxvf db-derby-10.4.2.0-bin.tar.gz
```

Figura 282: Instalación Hive Paso 10
Fuente: Elaboración propia.

11. Copiar los archivos extraídos en el paso 10, al directorio /usr/local/derby utilizando el siguiente comando en la terminal:

```
$ su -
passwd: (contraseña root)
# cd /home/user
# mv db-derby-10.4.2.0-bin /usr/local/derby
# exit
```

Figura 283: Instalación Hive Paso 11
Fuente: Elaboración propia.

12. Configurar el medio ambiente de Derby adicionando las siguientes líneas en el archivo ~/.bashrc:

```
export DERBY_HOME=/usr/local/derby
export PATH=$PATH:$DERBY_HOME/bin
Apache Hive
18
export CLASSPATH=$CLASSPATH:$DERBY_HOME/lib/derby.jar:$DERBY_HOME/lib/derbytools.jar
```

Figura 284: Instalación Hive Paso 12
Fuente: Elaboración propia.

13. A continuación, se debe aplicar los cambios en el sistema y ejecutarlo:

```
$ source ~/.bashrc
```

Figura 285: Instalación Hive Paso 13
Fuente: Elaboración propia.

14. Crear un directorio para almacenar el Metastore llamado $DERBY_HOME en el que se va a almacenar los datos. Se debe ejecutar el siguiente comando en la terminal:

```
$ mkdir $DERBY_HOME/data
```

Figura 286: Instalación Hive Paso 14
Fuente: Elaboración propia.

Con este último paso se ha completado la descarga, instalación y configuración del medio ambiente de Apache Derby.

Configuración del Metastore de Hive:

15. Con la configuración del Metastore se especifica a Hive la ubicación de la base de datos. Para eso es necesario la modificación del archivo hive-site.xml que está ubicado en el directorio $HIVE_HOME/conf. Por lo cual es necesario ejecutar el siguiente comando en la terminal:

```
$ cd $HIVE_HOME/conf
$ cp hive-default.xml.template hive-site.xml
```

Figura 287: Instalación Hive Paso 15
Fuente: Elaboración propia.

16. Editar el archivo hive-site.xml agregando la siguiente propiedad entre las etiquetas <configuration>, </configuration> como se lo muestra a continuación:

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:derby://localhost:1527/metastore_db;create=true </value>
  <description>JDBC connect string for a JDBC metastore </description>
</property>
```

Figura 288: Instalación Hive Paso 16
Fuente: Elaboración propia.

17. Se debe crear un archivo denominado jpox.properties y añadir las siguientes líneas:

```
javax.jdo.PersistenceManagerFactoryClass =

org.jpox.PersistenceManagerFactoryImpl
org.jpox.autoCreateSchema = false
org.jpox.validateTables = false
org.jpox.validateColumns = false
org.jpox.validateConstraints = false
org.jpox.storeManagerType = rdbms
org.jpox.autoCreateSchema = true
org.jpox.autoStartMechanismMode = checked
org.jpox.transactionIsolation = read_committed
javax.jdo.option.DetachAllOnCommit = true
javax.jdo.option.NontransactionalRead = true
javax.jdo.option.ConnectionDriverName = org.apache.derby.jdbc.ClientDriver
javax.jdo.option.ConnectionURL = jdbc:derby://hadoop1:1527/metastore_db;create = true
javax.jdo.option.ConnectionUserName = APP
javax.jdo.option.ConnectionPassword = mine
```

Figura 289: Instalación Hive Paso 17
Fuente: Elaboración propia.

Verificación de la instalación de Hive:

18. Antes de realizar la verificación de la instalación de Hive, en el HDFS es necesario crear una carpeta llamada tmp y otra llamada warehouse. A estas carpetas se les debe dar permisos de escritura, para lo cual se ejecutará el siguiente comando en la terminal del sistema:

```
$ $HADOOP_HOME/bin/hadoop fs -mkdir /tmp
$ $HADOOP_HOME/bin/hadoop fs -mkdir /user/hive/warehouse
$ $HADOOP_HOME/bin/hadoop fs -chmod g+w /tmp
$ $HADOOP_HOME/bin/hadoop fs -chmod g+w /user/hive/warehouse
```

Figura 290: Instalación Hive Paso 18
Fuente: Elaboración propia.

19. Para confirmar que Hive se encuentra instalado correctamente en el sistema se debe ejecutar el siguiente comando en la terminal:

```
$ cd $HIVE_HOME
$ bin/hive
```

Figura 291: Instalación Hive Paso 19.1
Fuente: Elaboración propia.

Se debe tener un resultado como el que se muestra a continuación:

```
Logging initialized using configuration in jar:file:/home/hadoop/hive-0.9.0/lib/hive-common-
0.9.0.jar!/hive-log4j.properties
Hive history file=/tmp/hadoop/hive_job_log_hadoop_201312121621_1494929084.txt
.....................
hive>
```

Figura 292: Instalación Hive Paso 19.2
Fuente: Elaboración propia.

Anexo 6: Instalación de Sqoop

La versión de Sqoop a instalar es la 1.4.6 y es necesario validar el sistema operativo en el que se va a ejecutar. Para Sqoop se recomienda el sistema operativo de Linux Ubuntu versión 14.04 LTS**,** ya que se lo pueda utilizar para el desarrollo y despliegue de aplicaciones.

La instalación de Sqoop1.4.6 según (White, 2015) se lo realiza de la siguiente manera:

Comprobación de la instalación de Java:

1. Se debe verificar si se tiene instalado en entorno Java en el sistema operativo haciendo uso del siguiente comando en la terminal:

```
$ java -version
```

Figura 293: Instalación Sqoop Paso 1.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

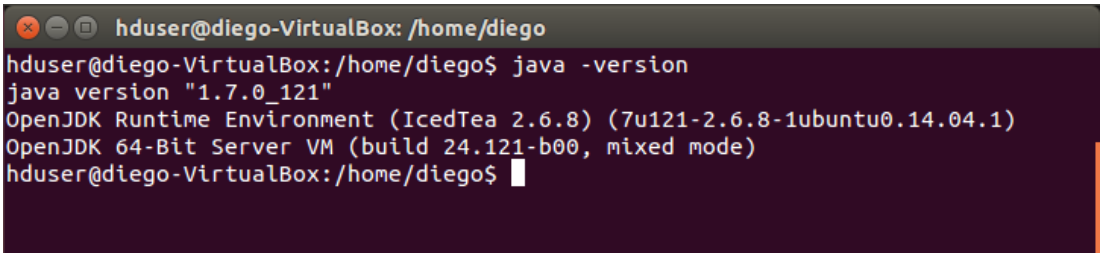


Figura 294: Instalación Sqoop Paso 1.2
Fuente: Elaboración propia.

En el caso de no tener instalado el entorno de Java seguir los pasos que se indican en el Anexo 2.

Comprobación de la instalación e inicio de Hadoop:

2. Se debe verificar la instalación Hadoop en el sistema haciendo uso del siguiente comando en la terminal del sistema:

```
$ hadoop version
```

Figura 295: Instalación Sqoop Paso 2.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

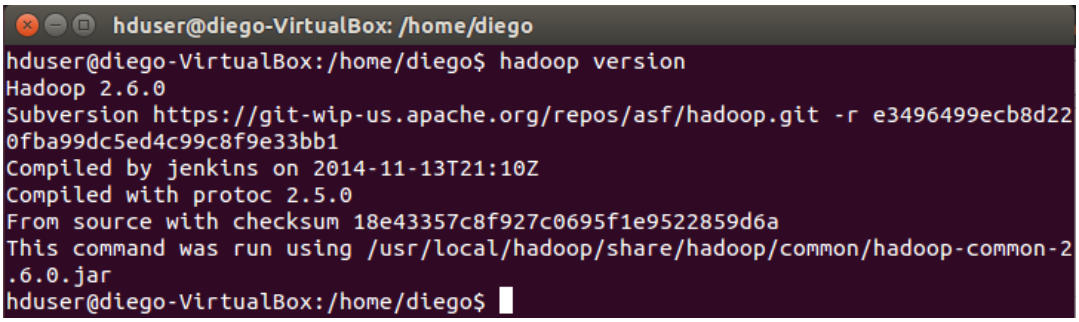


Figura 296: Instalación Sqoop Paso 2.2
Fuente: Elaboración propia.

Si está instalado correctamente Hadoop se debe iniciar el servicio como los puntos del subtema **Verificación de la instalación de Hadoop** en el Anexo 3, caso contrario seguir todos los pasos que conforman la instalación e inicio de Hadoop en el Anexo 3.

Instalación de Sqoop:

3. Se debe crear una carpeta con el nombre "work" en el directorio Home haciendo uso del siguiente comando en la terminal:

```
$ mkdir work
```

Figura 297: Instalación Sqoop Paso 3.1
Fuente: Elaboración propia

Al ejecutar el comando en la máquina virtual se observa que la carpeta se ha creado de la siguiente manera:

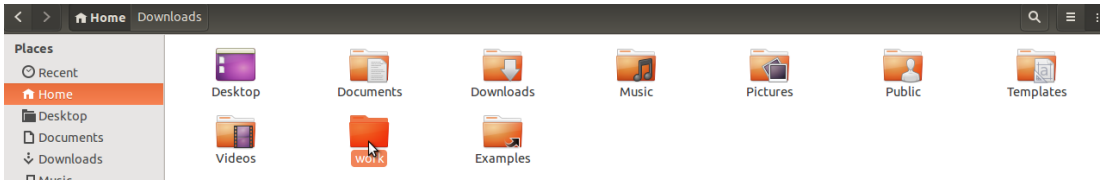


Figura 298: Instalación Sqoop Paso 3.2
Fuente: Elaboración propia

4. Se procede a descargar la versión de Sqoop 1.4.6 en la carpeta "work" creada en el paso 3, para lo cual ingresaremos al link http://www-eu.apache.org/dist/Sqoop/1.4.6/ y se debe seleccionar sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz. Como referencia se muestra la siguiente pantalla:

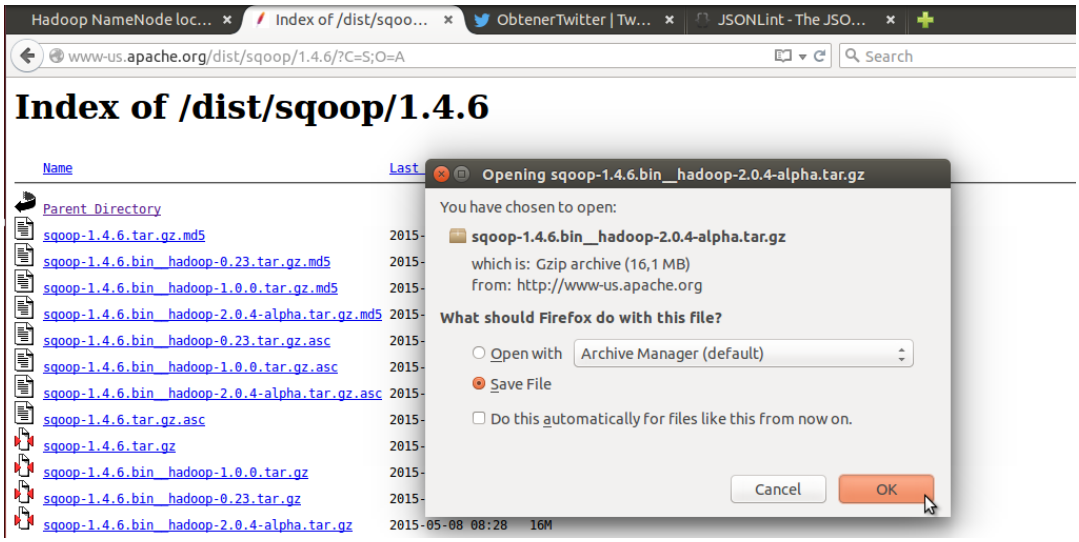


Figura 299: Instalación Sqoop Paso 4.1
Fuente: Elaboración propia

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "work" se ha descargado el archivo sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz como se lo indica de la siguiente manera:

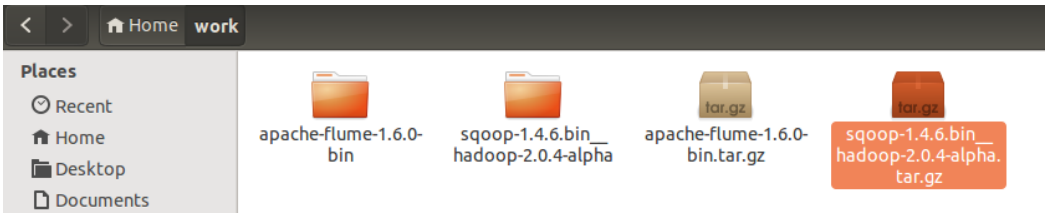


Figura 300: Instalación Sqoop Paso 4.2
Fuente: Elaboración propia

5. En la carpeta work se procede a extraer el archivo .tar de Sqoop 1.4.6 utilizando el siguiente comando en la terminal:

```
$ tar -xvzf sqoop-1.4.6.bin__hadoop-2.0.4-alpha.tar.gz
```

Figura 301: Instalación Sqoop Paso 5.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "work" se ha descomprimido el archivo y se ha creado la carpeta "sqoop-1.4.6.bin__hadoop-2.0.4-alpha" como se lo indica a continuación:

Figura 302: Instalación Sqoop Paso 5.2
Fuente: Elaboración propia.

6. A continuación, descargar el conector MySQL del link https://dev.mysql.com/downloads/connector/j/5.1.html y se debe seleccionar la versión 5.1.41, como se lo muestra en la siguiente pantalla:

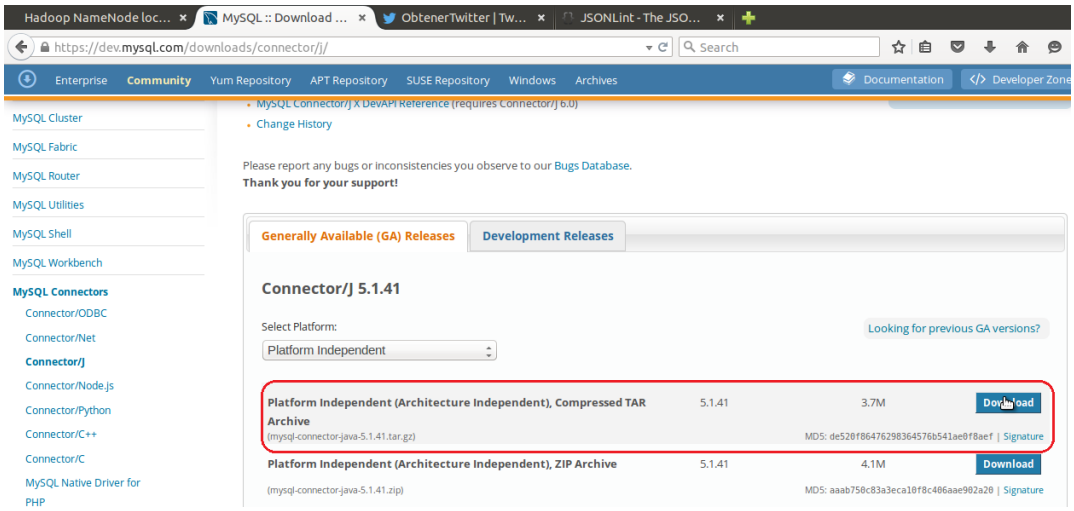


Figura 303: Instalación Sqoop Paso 6
Fuente: Elaboración propia.

7. En la máquina virtual se observa que en la carpeta "Downloads" se ha descargado el archivo MySQL-connector-java-5.1.41.tar.gz como se lo indica de la siguiente manera:

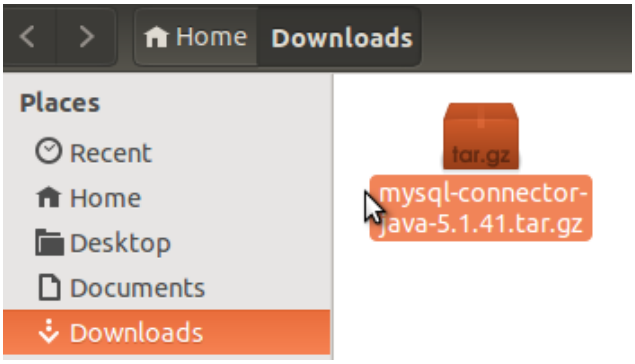


Figura 304: Instalación Sqoop Paso 7
Fuente: Elaboración propia

8. En la carpeta "Downloads" se procede a extraer el archivo .tar de MySQL-connector-java-5.1.41 utilizando el siguiente comando en la terminal:

```
$ tar -xvzf mysql-connector-java-5.1.41.tar.gz
```

Figura 305: Instalación Sqoop Paso 8.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "Downloads" se ha descomprimido el archivo y se ha creado la carpeta "MySQL-connector-java-5.1.41" como se lo indica a continuación:

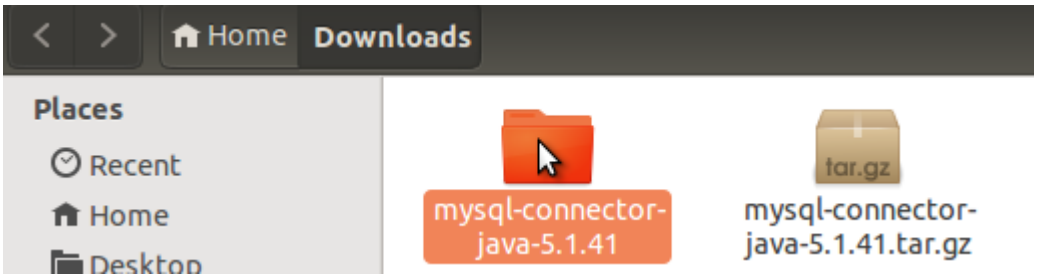


Figura 306: Instalación Sqoop Paso 8.2
Fuente: Elaboración propia.

9. Copiar el archivo "MySQL-connector-java-5.1.41-bin.jar" que se encuentra en la carpeta "MySQL-connector-java-5.1.41" de la carpeta "Downloads", hacia el directorio "lib" en la que se ubicó Sqoop de la carpeta "work" creada en el punto 5.

El archivo a mover se lo indica en la siguiente figura:

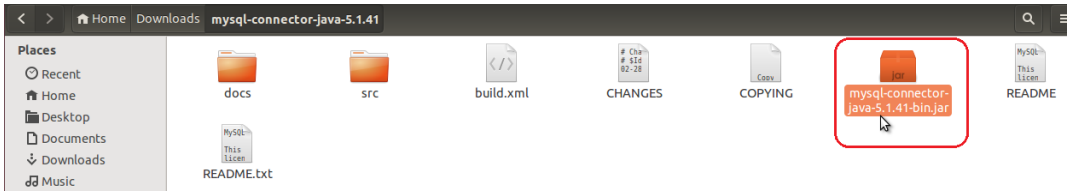


Figura 307: Instalación Sqoop Paso 9.1
Fuente: Elaboración propia.

En la máquina virtual se visualiza el archivo que es copiado al directorio especificado:

Figura 308: Instalación Sqoop Paso 9.2
Fuente: Elaboración propia.

10. A continuación, descargar el Oracle java conector del link http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html y se debe seleccionar el jar ojdbc6.jar, como se lo muestra en la siguiente pantalla:

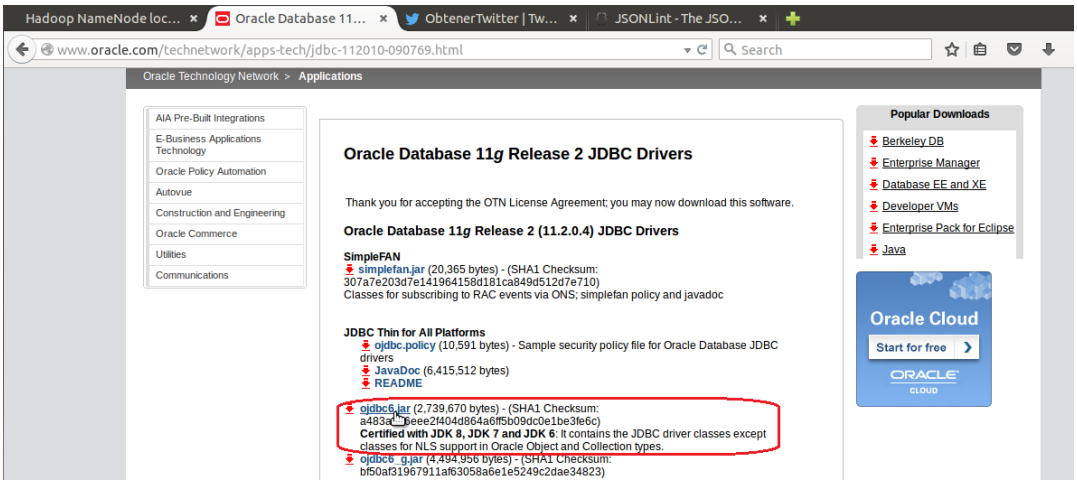


Figura 309: Instalación Sqoop Paso 10
Fuente: Elaboración propia.

11. En la máquina virtual se visualiza que en la carpeta "Downloads" se ha descargado el archivo ojdbc6.jar como se lo indica de la siguiente manera:

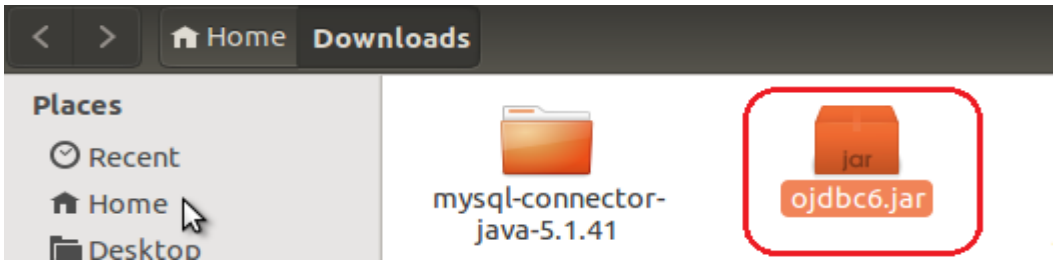


Figura 310: Instalación Sqoop Paso 11
Fuente: Elaboración propia

12. Copiar el archivo "ojdbc6.jar" que se encuentra en la carpeta "Downloads", hacia el directorio "lib" en la que se ubicó Sqoop de la carpeta "work" creada en el punto 5.

En la máquina virtual se visualiza el archivo que es copiado al directorio especificado:

Figura 311: Instalación Sqoop Paso 12
Fuente: Elaboración propia.

Configuración de Sqoop:

Para configurar Sqoop en nuestro sistema se debe modificar el archivo bashrc.

13. En el archivo .bashrc se debe configurar la carpeta de inicio, y las rutas de clase para Sqoop utilizando el siguiente comando en la terminal:

```
$ vim ~/.bashrc
```

Figura 312: Configuración de Sqoop Paso 13.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Sqoop 1.4.6. Las variables a agregar en la máquina virtual son:

```
export SQOOP_HOME=/home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha
export PATH=$SQOOP_HOME/bin:$PATH
```

Figura 313: Configuración de Sqoop Paso 13.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo bashrc de la máquina virtual quedó de la siguiente manera:

```
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin

export SQOOP_HOME=/home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha
export PATH=$SQOOP_HOME/bin:$PATH
```

Figura 314: Configuración de Sqoop Paso 13.3
Fuente: Elaboración propia.

14. Se debe aplicar los cambios realizados sobre el archivo bashrc ejecutando el siguiente comando en la terminal:

```
$ source ~/.bashrc
```

Figura 315: Instalación Sqoop Paso 14
Fuente: Elaboración propia.

15. En el directorio Home/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/conf existe el archivo sqoop-env.template.sh. Tal como se muestra a continuación:

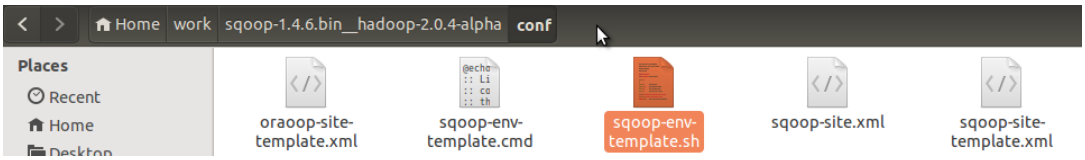


Figura 316: Instalación Sqoop Paso 15.1
Fuente: Elaboración propia.

De este archivo, se debe crear una copia de sqoop-env.template.sh, al cual se le debe asignar el nombre de sqoop-env.sh. Como referencia se muestra en la imagen:

Figura 317: Instalación Sqoop Paso 15.2
Fuente: Elaboración propia.

Se debe editar el archivo sqoop-env.sh, al cual se le debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Sqoop 1.4.6.

La configuración aplicada en el archivo sqoop-env.sh de la máquina virtual quedó de la siguiente manera:

```
#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/usr/local/hadoop

#Set path to where hadoop-*-core.jar is available
export HADOOP_MAPRED_HOME=/usr/local/hadoop

#set the path to where bin/hbase is available
#export HBASE_HOME=

#Set the path to where bin/hive is available
export HIVE_HOME=/usr/local/hive

#Set the path for where zookeper config dir is
#export ZOOCFGDIR=
```

Figura 318: Instalación Sqoop Paso 15.3
Fuente: Elaboración propia.

Verificación de Sqoop:

16. Para confirmar que Sqoop se encuentra instalado correctamente en el sistema se debe ejecutar el siguiente comando en la terminal del sistema:

```
$ sqoop version
```

Figura 319: Instalación Sqoop Paso 16.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se debe tener un resultado como el que se muestra a continuación:

```
File  Edit  View  Search  Terminal  Help
hduser@diego-VirtualBox:/home/diego$ sqoop version
Warning: /home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /home/diego/work/sqoop-1.4.6.bin__hadoop-2.0.4-alpha/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
17/04/16 17:18:28 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6
Sqoop 1.4.6
git commit id c0c5a81723759fa575844a0a1eae8f510fa32c25
Compiled by root on Mon Apr 27 14:38:36 CST 2015
hduser@diego-VirtualBox:/home/diego$
```

Figura 320: Instalación Sqoop Paso 16.2
Fuente: Elaboración propia.

Anexo 7: Instalación de Pig

La versión de Pig a instalar es la 0.16.0 y es necesario validar el sistema operativo en el que se va a ejecutar. Para Pig se recomienda el sistema operativo de Linux Ubuntu versión 14.04 LTS**,** ya que se lo puede utilizar para el desarrollo y despliegue de aplicaciones.

La instalación de Pig0.16.0 según (White, 2015) se lo realiza de la siguiente manera:

Comprobación de la instalación de Java:

1. Se debe verificar si se tiene instalado en entorno Java en el sistema operativo haciendo uso del siguiente comando en la terminal:

```
$ java -version
```

Figura 321: Instalación Pig Paso 1.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

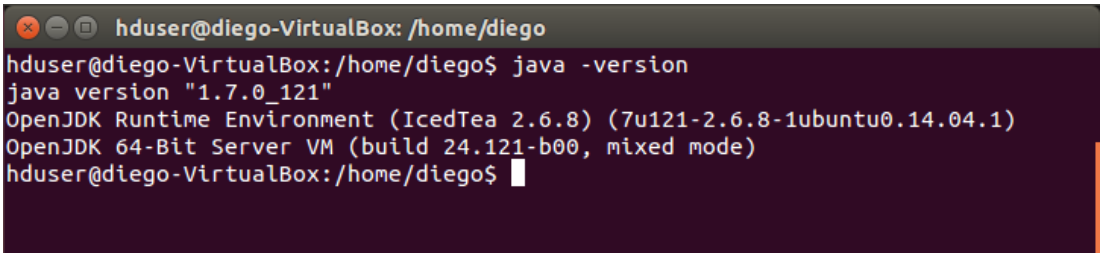


Figura 322: Instalación Pig Paso 1.2
Fuente: Elaboración propia.

En el caso de no tener instalado el entorno de Java seguir los pasos que se indican en el Anexo 2.

Comprobación de la instalación e inicio de Hadoop:

2. Se debe verificar la instalación Hadoop en el sistema haciendo uso del siguiente comando en la terminal del sistema:

```
$ hadoop version
```

Figura 323: Instalación Pig Paso 2.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se mostró de la siguiente manera:

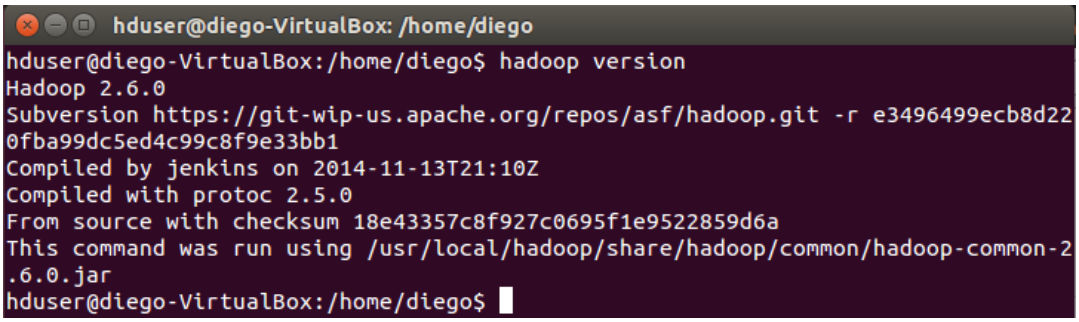


Figura 324: Instalación Pig Paso 2.2
Fuente: Elaboración propia.

Si está instalado correctamente Hadoop se debe iniciar el servicio como los puntos del subtema **Verificación de la instalación de Hadoop** en el Anexo 3, caso contrario seguir todos los pasos que conforman la instalación e inicio de Hadoop en el Anexo 3.

Instalación de Pig:

3. Se debe crear una carpeta con el nombre "Pig" en el directorio /usr/local/, haciendo uso del siguiente comando en la terminal(ingresar el password de súper usuario cuando lo solicite):

```
$ sudo mkdir /usr/local/pig
```

Figura 325: Instalación Pig Paso 3.1
Fuente: Elaboración propia

Al ejecutar el comando en la máquina virtual se observa que la carpeta se ha creado de la siguiente manera:

Figura 326: Instalación Pig Paso 3.2
Fuente: Elaboración propia

4. Se procede a descargar la versión de Pig-0.16.0 en la carpeta "Downloads", para lo cual ingresaremos al link http://www-us.apache.org/dist/pig/pig-0.16.0/ y se debe seleccionar pig-0.16.0.tar.gz. Como referencia se muestra la siguiente pantalla:

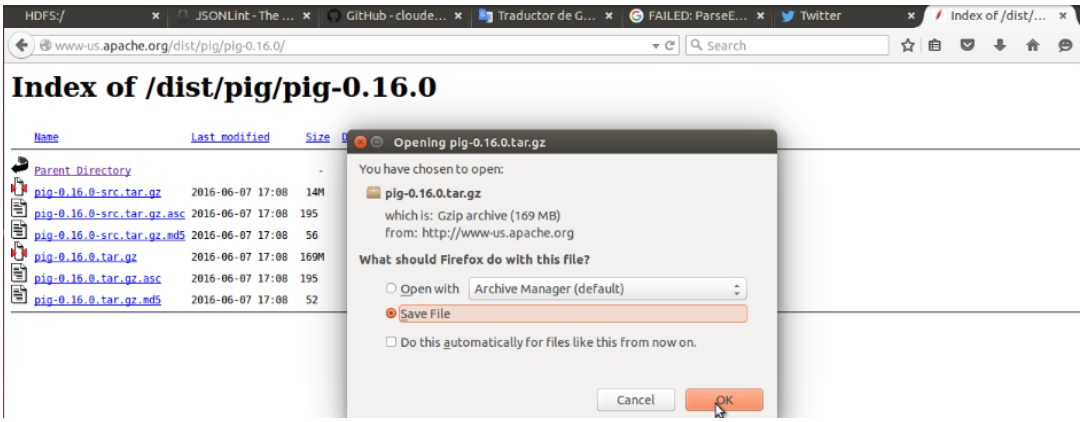


Figura 327: Instalación Pig Paso 4.1
Fuente: Elaboración propia

En la máquina virtual se observa que en la carpeta "Downloads" se ha descargado el archivo pig-0.16.0.tar.gz como se lo indica de la siguiente manera:

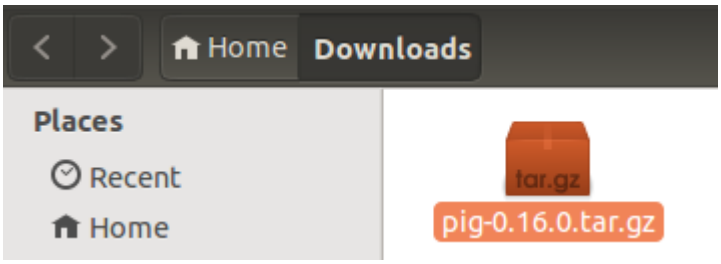


Figura 328: Instalación Pig Paso 4.2
Fuente: Elaboración propia

5. En la carpeta "Downloads" se procede a extraer el archivo .tar de pig-0.16.0 utilizando el siguiente comando en la terminal:

```
$ tar -xvzf pig-0.16.0.tar.gz
```

Figura 329: Instalación Pig Paso 5.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se observa que en la carpeta "Downloads" se ha descomprimido el archivo y se ha creado la carpeta "pig-0.16.0" como se lo indica a continuación:

Figura 330: Instalación Pig Paso 5.2
Fuente: Elaboración propia.

6. Mover todos los archivos que se encuentran en la carpeta "pig-0.16.0" hacia el directorio de la carpeta "Pig" creada en el punto 3.Se debe ejecutar el siguiente comando en la terminal(ingresar el password de súper usuario cuando lo solicite):

```
$ sudo mv pig-0.16.0/* /usr/local/pig
```

Figura 331: Instalación Pig Paso 6.1
Fuente: Elaboración propia.

En la máquina virtual se visualiza los archivos en el directorio especificado:

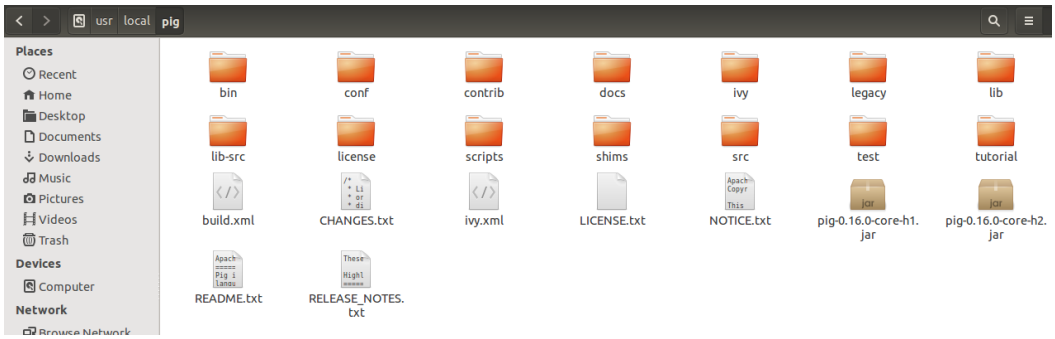


Figura 332: Instalación Pig Paso 6.2
Fuente: Elaboración propia.

Configuración de Pig:

Para configurar Pig en nuestro sistema se debe modificar el archivo bashrc.

7. En el archivo .bashrc se debe configurar la carpeta de inicio, y las rutas de clase para Pig utilizando el siguiente comando en la terminal:

```
$ vim ~/.bashrc
```

Figura 333: Configuración de Pig Paso 7.1
Fuente: Elaboración propia.

A continuación, se abrirá un archivo de texto en el que se debe agregar las variables de configuración de Java y Hadoop que se utilizarán para el funcionamiento de Pig 0.16.0. Las variables a agregar son:

```
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin
```

Figura 334: Configuración de Pig Paso 7.2
Fuente: Elaboración propia.

La configuración aplicada en el archivo bashrc de la máquina virtual quedó de la siguiente manera:

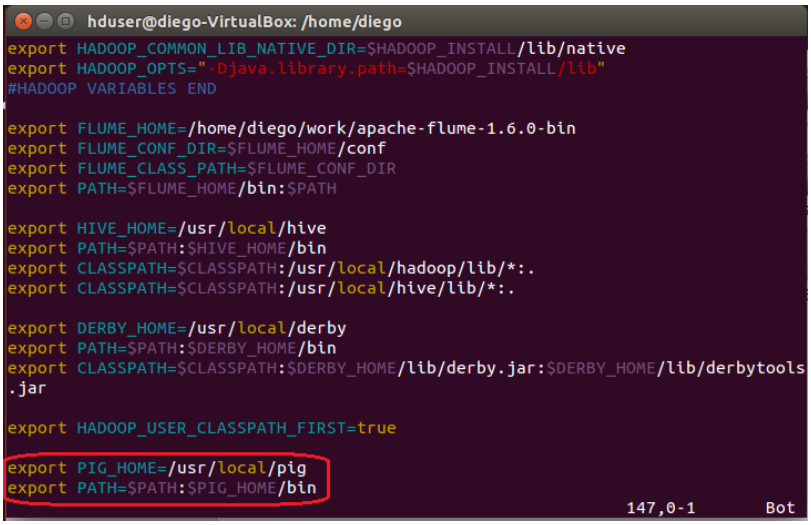


Figura 335: Configuración de Pig Paso 7.3
Fuente: Elaboración propia.

8. Se debe aplicar los cambios realizados sobre el archivo bashrc ejecutando el siguiente comando en la terminal:

```
$ source ~/.bashrc
```

Figura 336: Instalación Pig Paso 8.1
Fuente: Elaboración propia.

Verificación de Pig:

9. Para confirmar que Pig se encuentra instalado correctamente en el sistema se debe ejecutar el siguiente comando en la terminal del sistema:

```
$ pig
```

Figura 337: Instalación Pig Paso 9.1
Fuente: Elaboración propia.

Al ejecutar el comando en la máquina virtual se debe tener un resultado como el que se muestra a continuación:

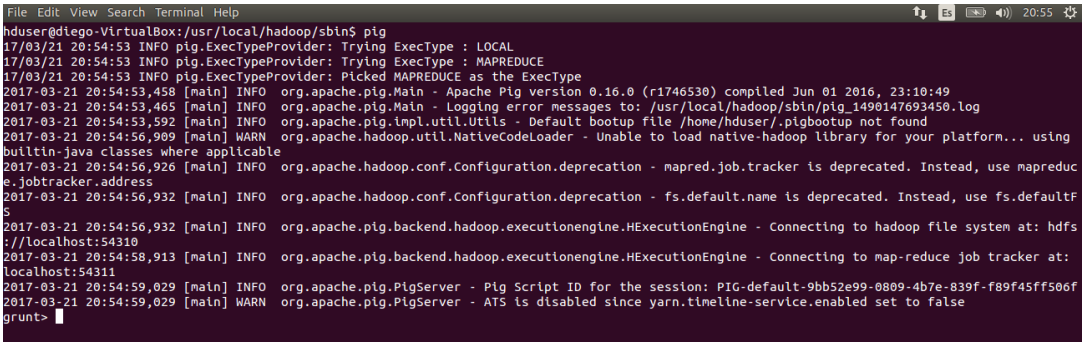


Figura 338: Instalación Pig Paso 9.2
Fuente: Elaboración propia.