Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

PROCEDIMIENTO DE DESCARGA E INDEXACIÓN DE TWEETS MEDIANTE MÉTODOS DE RECUPERACIÓN DE LA INFORMACIÓN

Trabajo de graduación previo a la obtención del título de Ingeniero en Sistemas y Telemática

Autores:
Pablo Zea Riofrío

Directores:

Ing. Marcos Orellana Cordero

Cuenca – Ecuador
2018

DEDICATORIA

Esta tesis la dedico a las personas que han influenciado en mi vida, brindando los mejores consejos, guiando y haciendo de mi una persona de bien.

Con todo mi amor y afecto a mi abuelita Rosario, mi hermana Lorena y mi madre Alicia

AGRADECIMIENTO

Mi agradecimiento especial a la Universidad del Azuay, que me abrió sus puertas para formarme profesionalmente.

Al Ingeniero Marcos Orellana por brindarme todos sus conocimientos y hacer posible el desarrollo de este artículo.

A mi abuela Rosario, mi hermana Lorena, mi madre Alicia y a toda mi familia que de alguna u otra manera me apoyaron.

Quiero expresarles a todos mis agradecimientos debido a que sin ustedes esto no hubiese sido posible, pero especialmente a Dios por brindarme una familia maravillosa y rodearme de personas estupendas que han contribuido en mi crecimiento personal y espiritual.

RESUMEN:

Se investiga un sistema para la obtención e indexación de datos de la red social Twitter basado en Hadoop y la plataforma de búsqueda Solr. Este sistema se diseñó e implementó para el manejo y búsqueda eficiente de vastas cantidades de datos, más en concreto de mensajes cortos o tweets, con la finalidad de alimentar un corpus de texto sobre el cual se pueda realizar operaciones de recuperación de información mediante la plataforma Solr. Hadoop es un framework de computación distribuida de código abierto que ha tomado popularidad en el manejo de big data, que en conjunto al manejo de índices de Solr resultan en un sistema con capacidades de almacenamiento y búsqueda de alto rendimiento.

Palabras clave: Recuperación de Información, Redes Sociales, Hadoop, Cluster

ABSTRACT

A system for obtaining and indexing data from the social network Twitter based on Hadoop and the search platform Solr was investigated. This system was designed and implemented for the efficient management and search of vast amounts of data, specifically short messages or tweets. It was intended to feed a text corpus to perform information retrieval operations through the Solr platform. Hadoop is an open source distributed computing framework that took popularity in big data management, which together with Solr index management, resulted in a system with high performance storage and search capabilities.

Keywords: Information Recovery, Social Networks, Hadoop, Cluster.

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by
Ing. Paul Arpi

ÍNDICE

Índice de tablas y figuras

Tablas

Figuras

Índice de anexos

CAPITULO I: IMPORTANCIA DEL ESTUDIO

1.1 INTRODUCCIÓN

La información es uno de los activos más valiosos con los que una organización o empresa puede contar, debido a que esta permite en la mayoría de los casos tomar decisiones acertadas (Desai, &Mehta, 2016). Poseer información oportuna y fiable puede involucrar una ventaja competitiva, siempre y cuando a esta convierta en conocimiento a través de un procesamiento adecuado como lo es el análisis de opiniones (Barnaghi, Ghaffari, & Breslin, 2016) o la identificación de eventos en tiempo real (Malas, &Vaidya, 2017). Muchas organizaciones dependen en su mayoría de datos generados por sí mismos, lo cual limita su conocimiento sobre variables externas que pueden influir al momento de la toma de decisiones o el desarrollo de un producto o servicio.

Una de las maneras más accesibles de obtener información es a través de redes sociales, debido a que la mayoría de estas generan información de acceso público, además se han extendido a tal punto que se han convertido en un canal dominante usado por los usuarios para denotar sus pensamientos, sentimientos y discusiones (Al-Hajjar, &Syed, 2015). Añadido a esto los avances en cuando a internet de banda ancha y a las aplicaciones móviles hacen posible el acceso de usuarios a estas plataformas desde prácticamente cualquier lugar (Jaywant, Shetty, &Musale, 2016).

Twitter es una de las redes sociales que más se han propagado en todo el mundo, y al contar con un formato de micro blog es la plataforma ideal para ser usada como fuente de información (Malas, &Vaidya, 2017). Ventajas de Twitter incluyen el acceso a datos de manera gratuita, así como información en constante actualización y su obtención mediante el uso de filtros específicos.

En este artículo se plantea la creación de un sistema, el cual sea capaz de adquirir e indexar una gran cantidad de datos de la red social Twitter en tiempo real (Big Data) (Dehdouh, Bentayeb, Boussaid, &Kabachi, 2014), así como la implementación de una arquitectura que tenga la capacidad de soportar la carga de operaciones de lectura, escritura, procesamiento y búsqueda de datos provocados al momento de la adquisición y recuperación de dicha información. Para lograr este objetivo se plantea el uso de herramientas para el manejo de Big Data como lo es Hadoop, sus medios de

almacenamiento HDFS y Hbase, el motor de búsqueda SolR, así también herramientas como Flume y Java que actúan como recolectores de información.

1.2 ESTADO DEL ARTE

La literatura habla de varias plataformas de redes sociales las cuales proveen métodos para la adquisición automática de información, como lo son Facebook, Twitter, Reddit, etc. Entre todas estas plataformas Twitter se encuentra como una de las más populares debido a la cantidad de usuarios activos, además de su facilidad y rapidez con la cual se notifican noticias y eventos (Lancieri, & Giovanetti, 2016). Esto se denota en el uso que varios autores le dan a esta red social para minería de datos (Disha, Sowmya, Chetan, & Seema, 2016), detección de eventos (Malas, & Vaidya, 2017), opinión mining (Barnaghi, Ghaffari, & Breslin, 2016) y análisis de sentimientos (Dudas, Weirman, & Griffin, 2017).

Twitter cuenta con varias formas de adquirir información, estas pueden ser mediante repositorio de datos, herramientas automatizadas provistas por terceros o las API oficiales de esta red social (Desai & Mehta, 2016). Cualquiera de los métodos mencionados es válido al momento de recolectar información, aunque hay que tomar en cuenta que los dos primeros están ligados a un sistema de alquiler dependiente de la información que sea consumida. Es evidente en la literatura la tendencia de los autores, al ser una práctica común optar por cualquiera de las API´s que Twitter facilita. Esta red social proporciona dos tipos de API´s: 1) La API Search permite obtener información filtrada mediante hashtags. (Desai, & Mehta, 2016); 2) La API Stream permite transmisión de datos en tiempo real (Desai, & Mehta, 2016). Varios autores mencionan que la API de Stream es la adecuada para temas como la minería de texto (Disha et al., 2016) e inclusive es la adecuada para generar corpus de texto masivos (Lucas et al., 2017).

La obtención de información de Twitter en tiempo real no es tema ajeno a la investigación, de hecho, debido a características como la generación de grandes cantidades de contenido y su constante actualización se habla de que gestor de base de datos o arquitectura es la predilecta para gestionar de manera adecuada esta información. En este tema, es concurrente la discusión acerca de los gestores de base de datos relacionales en contraste con las no relaciones, en cuanto a su rendimiento en

operaciones de lectura y escritura. Los resultados obtenidos en estas evaluaciones revelan entre otros que, en las bases de datos no relacionales el tamaño del set de datos no influye de gran manera en sus respuestas, siendo estas casi lineales, al contrario de un gestor de base de datos relacional (Schmid, Galicz, & Reinhardt, 2015). Una de las herramientas más populares entre los autores para gestionar información obtenida de Twitter es Hadoop, esto debido a su sistema de archivos distribuidos (HDFS) diseñado para el almacenamiento eficiente de volúmenes de datos masivos (Dehdouh, et al., 2014).

Hadoop es una herramienta que permite una gestión eficiente de Big Data, esto se debe principalmente a que este framework proporciona todo tipo de herramientas para la obtención, almacenamiento, análisis y visualización de información. Además, este ofrece flexibilidad al poder ser partícipe del intercambio de información con varias fuentes de datos, mediante herramientas como Flume (Selvan, & Moh, 2015) o mediante clientes programados los cuales actúan como receptores de información.

Uno de los objetivos principales en este artículo son los métodos de recuperación de información, esto hace referencia a recuperar datos no estructurados especialmente de documentos textuales (Greengrass, 2000). En este tema, la literatura habla de métodos de recuperación conceptualizados como es la indexación a través de la extracción de palabras clave (Hendez, & Achour, 2014), así como de herramientas ya concebidas y de alto rendimiento como son: Oracle Text y Lucene (Qian, & Wang, 2010). Lucene es una herramienta para indexación y búsqueda de texto basada en Java y que forma parte de dos proyectos de código abierto de Apache: 1) Haddop, 2) Lucene (Qian, & Wang, 2010). Esta herramienta es muy popular para la indexación en arquitecturas Hadoop, como por ejemplo en la recuperación de imágenes basadas en contenido (Gu, & Gao, 2012).

CAPITULO II: MARCO TEÓRICO

En el siguiente capítulo se describirán los conceptos necesarios para el entendimiento de este sistema como: recuperación de la información, Big Data, Hadoop, entre otros.

2.1 Recuperación de la Información

Este término también conocido como (IR) hace referencia a la recuperación de registros no estructurados, aquellos que se conforman principalmente de texto y se los conoce comúnmente como documentos. La recuperación de documentos a menudo se la suele llevar a cabo desde un repositorio organizado que se denomina como colección. La IR tiene como objetivo el encontrar documentos en una colección dada referentes a un tema específico. Estos documentos, deberán satisfacer la necesidad de información del usuario aquel que hizo la consulta.

(Greengrass, 2000).

2.2 Big Data

Big Data es una colección de grandes volúmenes de datos que no es posible procesarla por técnicas de computación tradicionales. Esto se debe a tres características: 1) Su volumen de datos; 2) La velocidad con la que se generan los datos; 3) Sus datos varían entre estructurados y no estructurados.

(Patil, & Phursule, 2014).

2.3 Hadoop

Hadoop es un framework de código abierto para gestionar Big Data que mediante el uso de un clúster de computadores, permite el procesamiento de grandes cantidades de datos (Patil, & Phursule, 2014). A continuación, se enunciarán los componentes principales de este framework.

2.3.1 Hadoop Distributed File System (HDFS)

HDFS es un sistema de archivos distribuidos diseñado para almacenar un gran volumen de datos. Este permite gestionar el almacenamiento en un clúster dividiendo los archivos entrantes en partes llamadas bloques y almacenándolos de manera

redundante en el conjunto de servidores. HDFS se compone de tres nodos principales los cuales son: 1) Nodo de nombres; 2) Nodo de datos; 3) Cliente HDFS.

1) Nodo de nombres: Actúa como maestro de los nodos de datos, por lo que contiene información acerca de todo el sistema de archivos de Hadoop. Su tarea principal es registrar metadatos, atributos y ubicaciones de bloques específicas en los nodos de datos.

2) Nodo de datos: Este nodo funciona como un nodo esclavo el cual es gestionado por el nodo de nombres. Sus principales tareas incluyen el almacenamiento de bloques como parte del HDFS y actuar como una plataforma para ejecutar procesos.

3) Cliente HDFS: Su principal función es el servir de enlace entre el nodo de nombres y los nodos de datos.

(Patil, & Phursule, 2014)

2.3.2 Hbase

Hbase una base de datos NoSQL que es usada comúnmente para almacenar masivas cantidades de datos no estructurados (Minanović, Gabelica, & Krstić, 2014). Esta base de datos se construye encima de HDFS y utiliza el paradigma orientado a columnas, esto permite tener un rendimiento y eficiencia adecuados al momento de trabajar con set de datos masivos (Patil, & Phursule, 2014).

2.3.3 Map Reduce

Este es un modelo de programación distribuida que usa Hadoop para el procesamiento y generación de grandes cantidades de datos. Map Reduce divide en varios fragmentos a los datos de entrada y se encarga asignar a cada uno de estos a una tarea de mapeado que permite procesar datos en paralelo (Patil, & Phursule, 2014).

2.3.4 Flume

Flume es un software que permite colectar datos de manera eficiente y fiable. Este es comúnmente usado para obtener y almacenar archivos logs de varios servidores en espacios como HDFS (Lai, Chen, Wu, Obaidat, 2014).

2.3.5 Hue

Hue es una interfaz web que se ejecuta sobre de Hadoop, permitiendo la visualización y gestión de HDFS, Hbase, además de brindar servicios para el procesamiento de datos orientados a ser amigables con los usuarios (Naik, 2017).

CAPITULO III: MÉTODO

3.1 Materiales y Métodos

En la siguiente sección se describirán y validarán métodos y herramientas seleccionadas para cumplir con tareas de extracción de datos de Twitter, el software utilizado para almacenar y procesar datos así también como la plataforma utilizada como motor de búsqueda y su método para realizar recuperación de la información.

3.1.1 Selección y Validación de Herramientas y Métodos

3.1.1.1 Gestión de Almacenamiento y Procesamiento de Datos

Un aspecto de gran importancia en este artículo fue la gestión de almacenamiento y procesamiento de datos, siendo esto un problema severo por el degrado que suelen tener las bases de datos convencionales, al aumentar el tamaño del set de datos que manejan (Schmid, Galicz, & Reinhardt, 2015). Por este motivo hubo la necesidad de usar el software y estructura más adecuado para gestionar el volumen de datos esperados por el sistema. Esta selección se realizó mediante fundamentos hallados dentro de la literatura revisada, autores como Patil et al., (2014), Lai et al., (2014), Selvan et al., (2015), entre otros mencionan en sus trabajos el impacto en cuanto al rendimiento de Hadoop al momento de gestionar de Big Data, específicamente en información de medios sociales.

Antes de empezar con la implementación de Hadoop fue necesario comparar y seleccionar una distribución de este framework, esto se realizó mediante el cuadrante mágico de Gartner. Un criterio importante para su selección fue que esta herramienta sea de distribución libre u *open source*, lo cual redujo los candidatos a dos contendientes principales: Cloudera y Hortonworks, cuya comparación se puede observar en la ilustración 1:

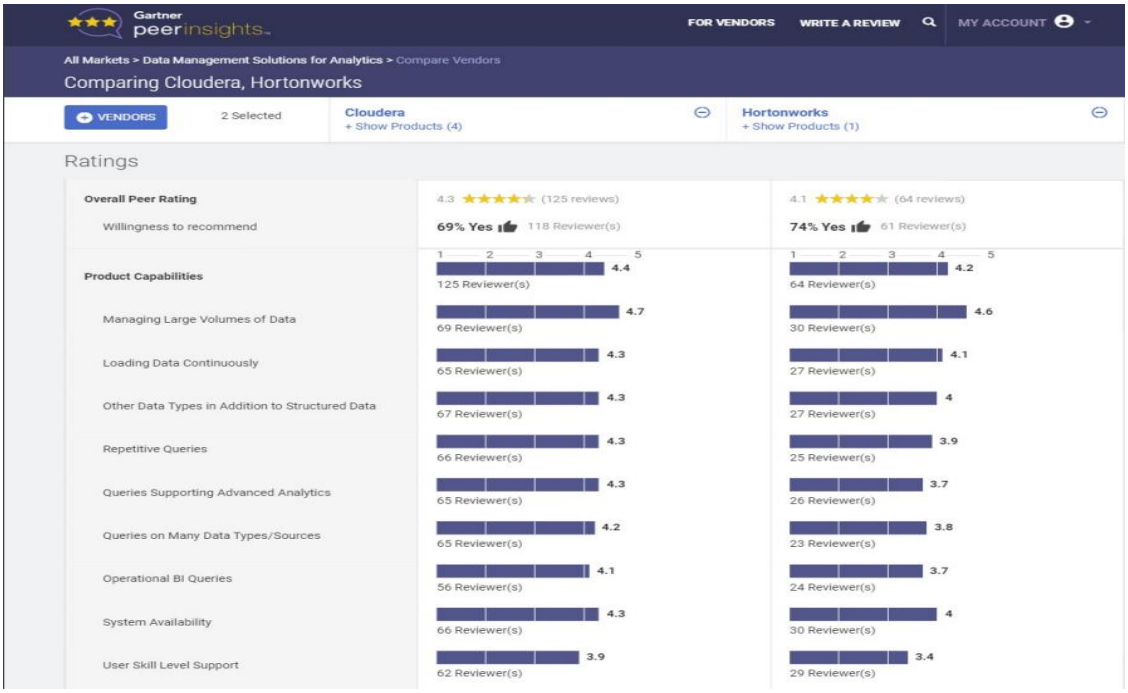


Ilustración 1 Comparación de Distribuciones de Hadoop tomado de https://www.gartner.com/reviews/market/data-warehouse-solutions/compare/cloudera-vs-hortonworks-vs-amazon-web-services

De la ilustración 1 podemos exponer que las dos distribuciones son muy parecidas en cuanto a sus características de rendimiento en gestión de datos, pero un número mayor de usuarios han preferido la distribución Hadoop de Cloudera.

3.1.1.2 Método de Indexación y Recuperación de Información

La literatura revisada nos habla del uso de la librería Lucene como método de recuperación de información, pues esta librería permite realizar operaciones de indexación y búsqueda de datos. Para usar esta librería Hadoop ofrece plataformas de búsqueda basadas en Lucene como: ElasticSeacrh y Solr (Lancieri, &Giovanetti, 2016).

Para la elección de una plataforma de búsqueda basada en Lucene, se decidió por la distribución de Hadoop. Este fue el punto de partida para la implementación de un motor de búsqueda. Por esta razón la plataforma implementada fue Solr, aquella que Cloudera incluye en sus paquetes de instalación.

3.1.1.3 Métodos de Extracción de Datos Twitter

Existen varios métodos para extraer datos de esta red social, en la sección de estado del arte se ha mencionado el porqué de la popularidad de la API de Twitter para

este fin, pero esto no es suficiente para completar con su función debido a que se necesita una plataforma que soporte el uso de esta API.

La selección de esta plataforma se realizó mediante criterios literarios de rendimiento y sobre todo que se apeguen al *framework* para manejo de datos seleccionado, en este caso Hadoop.

En la literatura autores como Santos et al., (2016), Cunha et al., (2015) y Azziki et al., (2016), mencionan utilizar la librería Twitter4J y el lenguaje de programación Java como medio para obtener datos de Twitter y su posterior gestión en Hadoop, esto se debe al acoplamiento y facilidades que brinda este *framework* con Java al estar escrito en este lenguaje.

El manejo directo de la API de Twitter hacia Hadoop es posible como menciona Selvan, &Moh, (2015),quién habla de un servicio provisto por este *framework* denominado Flume, aquel que puede trabajar con una fuente de datos de cualquier medio incluido Twitter y almacenar datos a manera de *stream* en el HDFS de Hadoop.

La elección de la plataforma para extraer de datos de Twitter estaría normalmente inclinada al uso de Flume, pero esto no fue así debido a la manera que este trabaja, almacenando directamente los tweets en el HDFS y obligando a realizar indexación por lotes de manera manual.

El proceso de funcionamiento pensado para este sistema consta de extraer, indexar y almacenar en tiempo real, por este motivo utilizar Java como plataforma de programación es lo ideal. Java permite crear un cliente Solr al que se envían los tweets a indexar, para luego almacenarlos en conjunto con sus índices en el HDFS, revirtiendo el proceso realizado por Flume y evitando así la escritura de los tweets y sus índices en tiempos diferentes.

3.1.2 Recursos Hardware

Montar una arquitectura Hadoop destinada a producción requiere de grandes recursos hardware, como señala la documentación de Cloudera en donde se definen requisitos mínimos de al menos 10 nodos por clúster con una capacidad de disco que va desde 4 a 6 TB, y memoria RAM de 64 a 128 GB (O' Dell, 2013).Por razones de experimentación se montó un Hadoop con 4 nodos, aquellos que trabajan como

Cloudera Manager, Nodo de nombre, Nodo Secundario y Nodo de datos. A continuación, se presentan las características de los recursos mencionados.

Tabla 1 Recursos para implementar Hadoop

Marca	Modelo	Nombre de Host	RAM	Disco Duro	Rol	Nombre de Servidor
ASUS	Optiplex 750	host1.uda.com	8 GB	500 TGB	Cloudera Manager	Servidor 1
ASUS	Inspiron 5567 SignatureEditi on	host3.uda.com	16 GB	1TB	Nodo Primario	Servidor 2
ASUS	Inspiron 5567 SignatureEditi on	host2.uda.com	16 GB	1 TB	Nodo Secundario	Servidor 3
ASUS	X751L	host4.uda.com	8 GB	800 GB	Nodo de datos	Servidor 4

Los recursos enlistados en la tabla 1 no fueron utilizados en su totalidad debido a que estos equipos pertenecen al laboratorio de investigación y desarrollo en informática (LIDI)de la Universidad del Azuay. Por esta razón se optó por usar virtualización a través del software Oracle Virtual Box. En la siguiente tabla podemos observar los recursos utilizados para la virtualización.

Tabla 2 Recursos utilizados para virtualización

Nombre de Host	Sistema Operativo Virtualizado	Tipo de instalación	Dirección Ip	sobre el que se virtualizó
host1.uda.com	Centos 7	Escritorio Gnome	10.10.220.51	Servidor 1
host2.uda.com	Centos 7	Servidor para procesamiento	10.10.220.52	Servidor 2
host3.uda.com	Centos 7	Servidor para procesamiento	10.10.220.53	Servidor 3
host4.uda.com	Centos 7	Servidor para procesamiento	10.10.220.54	Servidor 4

3.2 Método

3.2.1 Arquitectura del Sistema

La arquitectura del sistema planteado consta de cuatro componentes principales: 1) Hadoop; 2) Extractor de datos; 3) Indexador de datos; 4) Consultor de datos.

1) Hadoop: Framework de computación distribuida que soporta el almacenamiento, indexación y consulta de datos.

2) Extractor de datos: Componente que obtiene datos provenientes de Twitter a manera de *streaming*.
3) Indexador de datos: Crea índices sobre los datos para facilitar operaciones de recuperación de información.
4) Consultor de datos: Permite realizar operaciones de recuperación de información.

En la siguiente figura se puede observar las interacciones de cada uno de los componentes:

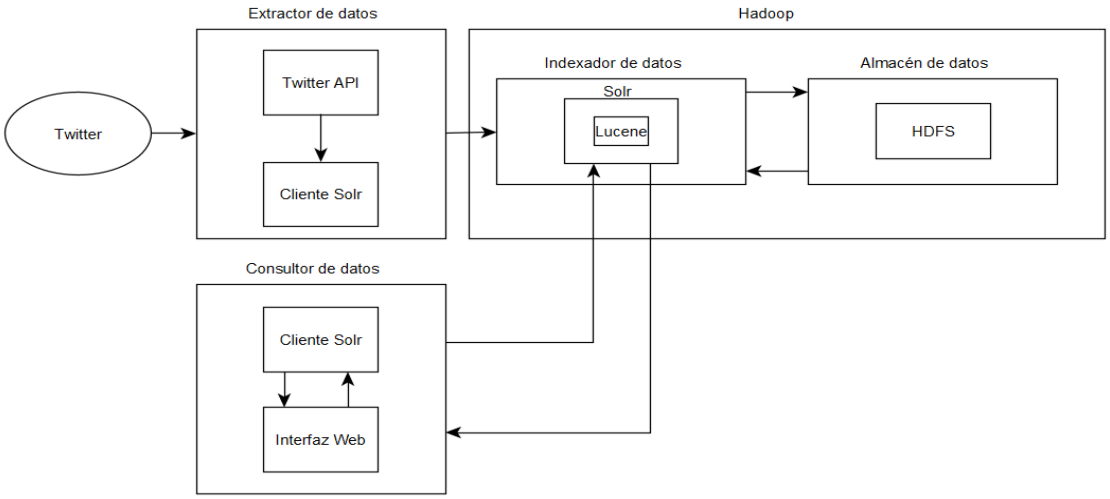


Ilustración 2 Estructura del sistema de Extracción e indexación de Tweets

Donde el extractor de datos necesita autentificarse con Twitter a través su API antes de empezar a obtener datos, realizado esto la API dirige todo el tráfico de información de Twitter hacia el extractor controlando que no se sobrepase el límite de tweets denominado *GardenHose*. El extractor de datos filtra la información del tweet que necesita descartando lo demás y genera una colección de documentos listos para indexar y almacenar.

La tarea de indexación involucra comunicar el cliente Solr con el indexador de datos en busca de disponibilidad del *core* de la plataforma Solr aquel que creará los índices. Comprobada esta se envía la colección de documentos hacia el indexador de datos que usa a Lucene para indexar estos documentos y almacenarlos en el HDFS de Hadoop.

Los procesos de recuperación de información los realiza el consultor de datos, este usa de una interfaz web para obtener parámetros de búsqueda, aquellos que

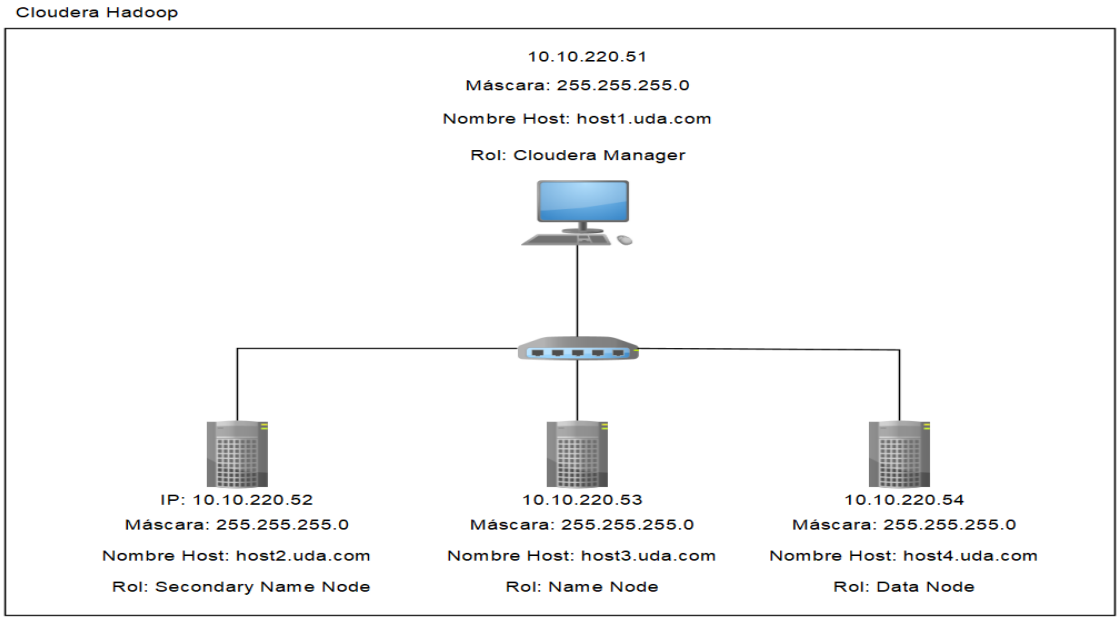
mediante un cliente Solr conocido como Ajax-Solr hacen peticiones o query´s a Solr. El motor de búsqueda Solr utilizará sus índices para buscar términos similares dentro del HDFS y los retornará hacia el cliente para su visualización.

3.2.1.1 Hadoop

Es la columna vertebral del sistema debido a que todos los módulos necesitan interactuar con este para cumplir sus funciones. Este *framework* de computación brinda altos rendimientos en procesamiento de datos, así también en operaciones de lectura y escritura debido a su naturaleza distribuida y uso de herramientas como el software Map Reduce y el sistema de archivos distribuidos HDFS.

Hadoop cuenta con tres elementos específicos para completar el sistema: 1) Sistema de archivos distribuidos 'HDFS'; 2) Framework de indexación 'Solr'; 3) Procesamiento distribuido 'Map Reduce'.

A continuación, se muestra el esquema de red utilizado para la implementación de Hadoop:



Para más información sobre la instalación y configuración de Hadoop revisar el Anexo 1.1 Implementación de Hadoop.

3.2.1.2 Indexador de Datos

Se encarga de crear índices a partir de documentos provenientes de Twitter, con la finalidad de mejorar el rendimiento de búsquedas en base a términos encontrados en estos documentos. Este componente utiliza la plataforma de búsqueda Solr que tiene a Lucene como *core* para realizar los procesos de indexación.

Lucene es una librería de código abierto para recuperación de información comúnmente usada para construir de motores de búsqueda (Gu, & Gao, 2012).

Este componente funciona como plataforma de búsqueda de información para Hadoop, su principal función consiste en agilitar tiempos de respuestas en operaciones de búsqueda, mediante índices generados a partir de archivos.

El proceso de indexación de archivos se realiza por lotes, donde el extractor de datos arma colecciones de 2000 tweets y los envía a Solr a generar y almacenar índices a partir de estos. Los datos indexados se almacenan en HDFS en tiempo real con formato de matriz inversa, al cual el indexador convierte todos los archivos.

El framework Solr en Hadoop puede ser implementado mediante la agregación de un servicio en el cliente de Cloudera Manager. La generación de índices en la plataforma Solr necesita de la creación de un *core* configurado de acuerdo a los datos que serán indexados. Para más información sobre la instalación y configuración de Solr revisar el Anexo 1.2 Implementación del Indexador de Datos.

3.2.1.3 Extractor de Datos

Este componente intermedia información entre Twitter y el HDFS de Haddop, su función principal consiste en el obtener tweets en tiempo real, filtrar información importante, convertirlos a un formato JSON y enviarlos al indexador de datos.

La extracción de datos de Twitter se realizó por medio de la API de *stream* y el lenguaje de programación Java, esto debido a 3 factores importantes: 1)El acoplamiento que tiene este con Hadoop (Patil, & Phursule, 2014); 2)Facilitan la creación y gestión de clientes para componentes de Hadoop como HDFS y Solr, a través de librerías especializadas; 3) Lo estipulado por la literatura aquella que resalta a autores como Cunha, et al., (2015), Al-Hajjar, et al., (2015), Azziki, et al., (2016), aquellos que hacen uso de estas herramientas mencionadas..

Debido a condiciones especiales presentadas por el cliente Solr, fue necesario el uso de la herramienta Maven para realizar la compilación del proyecto. Esto se debe a que el cliente Solr necesita implementar dependencias para hacer uso de sentencias propias de este cliente.

Maven es una herramienta para el manejo de proyectos software basada en el concepto de modelo de proyecto (POM). En esta herramienta se especifican dependencias mediante la creación de un archivo Pom.xml, donde se enuncian los paquetes necesarios por la aplicación Java para cumplir sus funciones. Para más información sobre la creación del extractor de datos revisar el Anexo 1.3 Implementación del Extractor de Datos.

3.2.1.3.1 Formato de Datos

Los tweets obtenidos del extractor de datos se estructuran en un formato JSON, brindándoles etiquetas y organizando sus datos facilitando así la identificación de cada uno de los elementos que contiene un tweet. A continuación, se presentan el formato de estos datos una vez almacenados en el HDFS:

1) id_tweet: Identificador único de un tweet seleccionado por Twitter.
2) usu_tweet: Usuario al cual le pertenece el tweet.
3) cont_tweet: Contenido del tweet.
4) usu_id_tweet: Identificador único del usuario seleccionado por Twitter.
5) des_usu_tweet: Descripción del usuario.
6) cod_pais_tweet Código del país al cuál el usuario pertenece.
7) fuent_tweet: Plataforma sobre la cual se realizó el tweet.
8) geol_acti_tweet: Indica si el usuario tiene activada la geolocalización de los tweets.
9) cant_amig_tweet: Indica la cantidad de amigos con los que cuenta el usuario.
10) loca_tweet: Muestra las coordenadas de donde se realizó el tweet.
11) luga_tweet: Muestra la ciudad de donde se realizó el tweet.
12) leng_tweet: Muestra el lenguaje en el cual está escrito el tweet.

3.2.1.4 Consultor de Datos

Recupera información asociada a una cadena de búsqueda definida por un usuario, haciendo uso de índices creados a priori. Este componente utiliza el motor de búsqueda de Solr para consultar la información almacenada en el HDFS.

El consultor de datos se lo concibe como un cliente programable, que tiene acceso a Solr mediante credenciales asociadas a este elemento. Este consiste de un cliente web construido con AJAX Solr. Para más información sobre la creación del consultor de datos revisar el Anexo 1.4 Implementación del Consultor de Datos.

CAPÍTULO IV: RESULTADOS

4.1 Resultados

Los experimentos realizados sobre los elementos que conforman este sistema demuestran la necesidad de implementar un *framework* de computación distribuida como lo es Hadoop para gestionar información obtenida de redes sociales, además de corroborar con lo enunciado por autores como Azziki et al., (2016) quien afirma la dificultad del manejo de datos no estructurados y su trato imperativo como Big Data.

Este sistema es capaz de extraer alrededor de 723,867 tweets en un lapso de cuatro horas, lo que equivale a 300 MB de datos. Añadido a esto el tamaño de los archivos almacenados en HDFS es incluso más pesado debido a la creación de índices de Solr. Esto demuestra la gran densidad de los datos de Twitter, además de la velocidad a la que estos se propagan convirtiéndose en un problema a la hora de gestionarlos sin el manejo de índices.

En la ilustración 12 se puede apreciar el comportamiento de las operaciones de escritura en el HDFS. Estas operaciones adoptan un comportamiento peculiar de picos altos y bajos debido a la forma en que se manejan los archivos a indexar, tomando colecciones de 2,000 tweets para la creación de índices y luego almacenarlos por lotes. Por este motivo las operaciones de escritura de archivos disminuyen mientras se preparan e indexan las colecciones de tweets.

Por otro lado, factores como la velocidad de transferencia de la red y el tamaño de las colecciones de tweets son otro factor que afecta al comportamiento de escritura, esto debido a que las colecciones de tweets con textos pobres en contenido son transmitidos

e indexados mucho más rápido necesitando de menos recursos de procesamiento, HDFS y red.

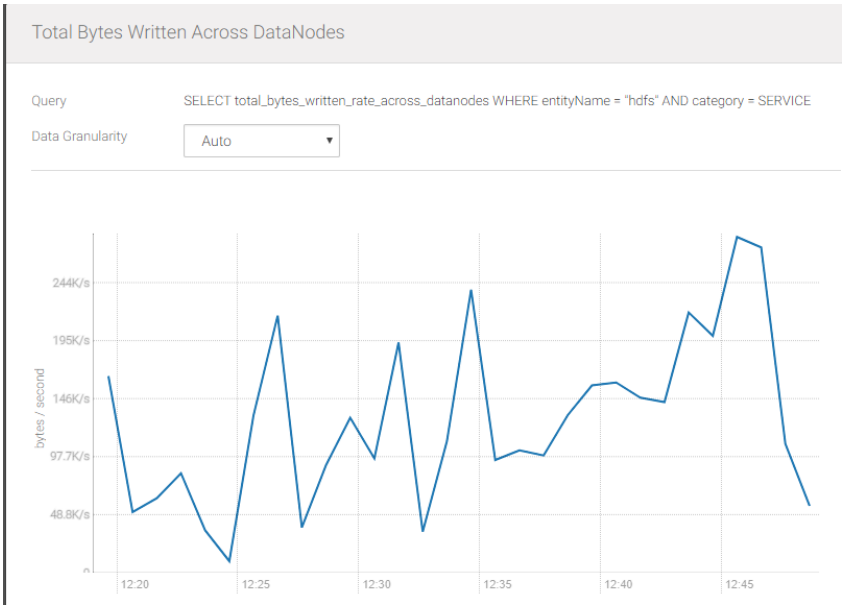


Ilustración 4 Total de bytes escritos en los nodos de datos

Los datos obtenidos de twitter se recuperan a manera de *streaming*, es decir en tiempo real, por ende, la indexación y almacenamiento de datos también debe ser manejada del mismo modo. Esto significa que tareas de escritura son constantes y consumen una gran cantidad de recursos, añadido a esto la recuperación de información requiere de la lectura de una gran parte de archivos almacenados como lo son índices de Solr y archivos de tweets relacionados con operaciones de búsqueda. En la ilustración 13 se observa el total de *bytes* escritos en el disco del clúster representado por el polígono verde, mientras el total de *bytes* leídos en el disco se representa con azul. De esta figura podemos corroborar el alto consumo de recursos de disco debido a la constante lectura y escritura de índices y tweets, además es clara la alta densidad de los datos obtenidos al necesitar picos de hasta siete *Megabytes* por segundo en estas operaciones.

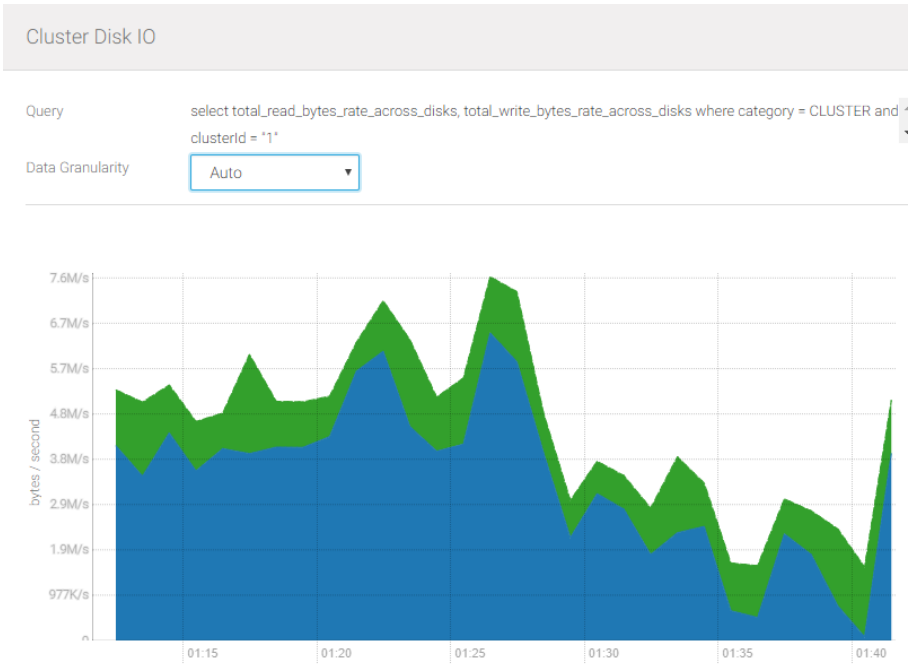


Ilustración 5Operaciones de entrada y salida en el disco del clúster

El consumo de recursos es notable en cuanto a procesos que involucran operaciones del disco del clúster, por otro lado, si lo comparamos con el procesamiento computacional necesario para conllevar todas las operaciones que realiza el sistema, además del control de los componentes que lo conforman, podremos observar que apenas hemos tocado la superficie de la capacidad total del Hadoop disponible. Esto se puede apreciar en la ilustración 14 donde se enseña el porcentaje promedio de utilización del CPU del clúster aquel que señala el uso de un diez por ciento total de su capacidad.

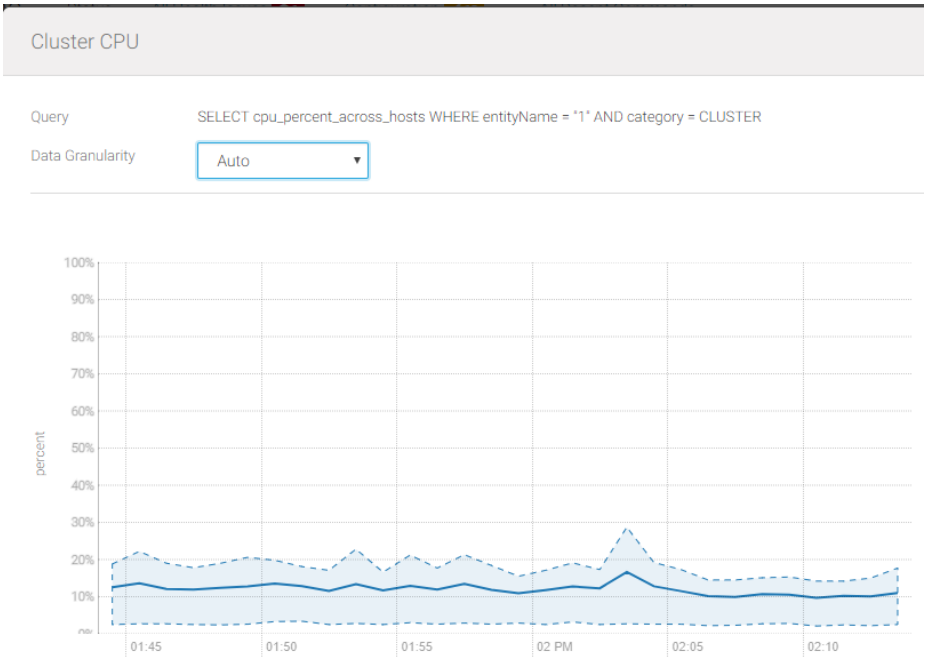


Ilustración 6 Porcentaje de utilización de CPU del Clúster

La recuperación de información asociada a tweets involucra a menudo el retorno de grandes cantidades de datos, esto debido al gran tamaño del set de datos sobre el que se realizan las funciones de búsqueda. Esto se puede denotar en la siguiente ilustración en la cual se refleja el consumo de recursos al momento de realizar una operación de búsqueda de tweets, donde el polígono azul índica la cantidad de *bytes* leídos en el HDFS para satisfacer esta operación de búsqueda, mientras el polígono verde indica la escritura de *bytes* realizada por el almacenamiento de tweets. El comportamiento observado en esta figura demuestra la tendencia de todas las operaciones de recuperación de información aquellas que disparan un pico de lectura de datos cada vez que Solr procesa una petición.

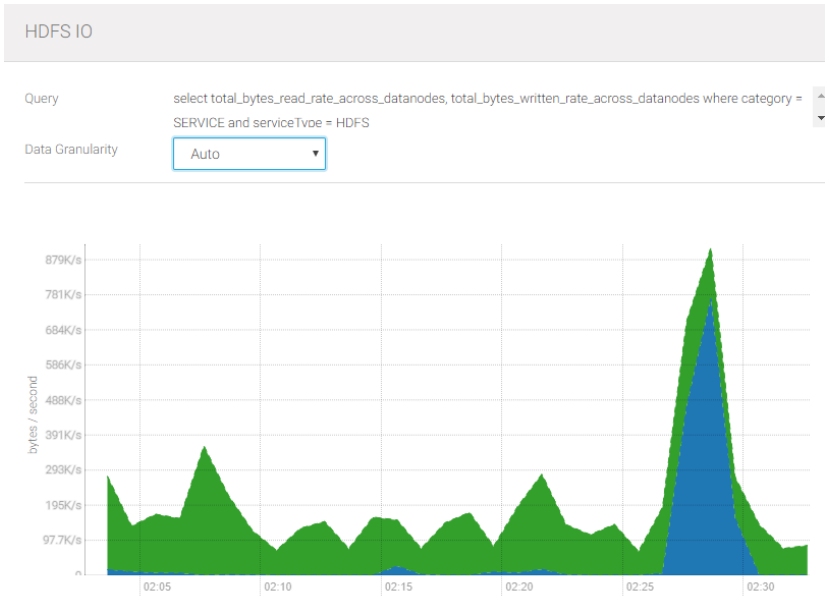


Ilustración 7 Datos leídos en HDFS al momento de una operación de recuperación de la información

El sistema es capaz de recuperar datos de una manera precisa y eficaz a través del uso de índices. Estos son la razón de la eliminación de búsquedas directas sobre el HDFS, en su lugar se accede a estos índices para localizar los tweets pertenecientes a un criterio de búsqueda. En la siguiente ilustración se enseñan los parámetros necesarios para recuperar información relacionada con la palabra ¨ECUADOR¨, del campo de contenido de tweets denominado como ¨cont_tweet¨, sobre el *core* de indexación llamado ¨indexacion_tweets¨

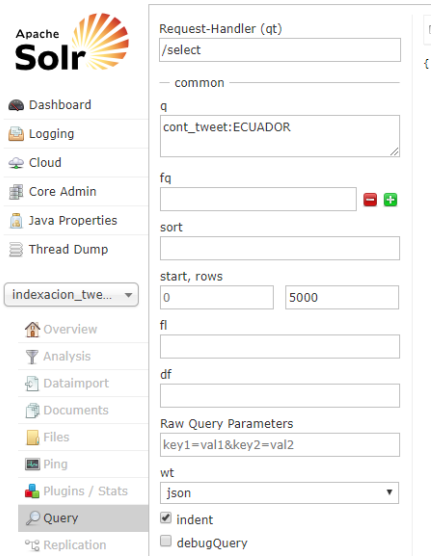


Ilustración 8 Parámetros de Solr para realizar un query

El resultado del *query* realizado con los parámetros mencionados en la anterior figura se muestra en la ilustración 9, donde el objeto *responseHeader* nos muestra información sobre el estado de la consulta el cual incluye la cadena de búsqueda, número de documentos y el formato en el cual se recuperaron los documentos. Mientras que el objeto response nos muestra el contenido de los documentos recuperados.


```
{
  "responseHeader": {
    "status": 0,
    "QTime": 16,
    "params": {
      "indent": "true",
      "q": "cont_tweet:ECUADOR",
      "_": "1536875597508",
      "wt": "json",
      "rows": "5000"
    }
  },
  "response": {
    "numFound": 3569,
    "start": 0,
    "docs": [
      {
        "id_tweet": " 1007405702204227584",
        "usu_tweet": " @SegundoEnfoque",
        "cont_tweet": " #Ecuador ??",
        "usu_id_tweet": " 3094944120",
        "des_usu_tweet": " SegundoEnfoque es un medio online que se edita desde Buenos Aires. SegundoEnfoque es información confiable y per
        "stat_usu_tweet": " null",
        "cod_pais_tweet": "null",
        "fuent_tweet": " <a href=https://about.twitter.com/products/tweetdeck rel=nofollow>TweetDeck</a>",
        "retw_byme_tweet": " false",
        "enti_hash_tweet": " [Ltwitter4j.HashtagEntity@46486c",
        "retw_tweet": " false",
        "geol_acti_tweet": " false",
        "enti_menc_tweet": " [Ltwitter4j.UserMentionEntity@1dd6cf8",
        "cant_amig_tweet": 5127,
        "cant_segi_tweet": 51531,
        "enti_url_tweet": " [Ltwitter4j.URLEntity@1e76c42",
        "fech_crea_tweet": " Thu Jun 14 18:33:30 COT 2018",
        "loca_tweet": " Argentina",
        "nume_favo_tweet": 0,
        "cont_retw_tweet": 0,
```


Ilustración 9Resultado de la recuperación de la información

El rendimiento en cuanto a operaciones de recuperación de información es notable, debido a la capacidad del consultor de datos de recuperar un set de datos conformado por 375,000 tweets en cuestión de segundos, además de ofrecer funciones como el filtrado de resultados dependiendo de las búsquedas.

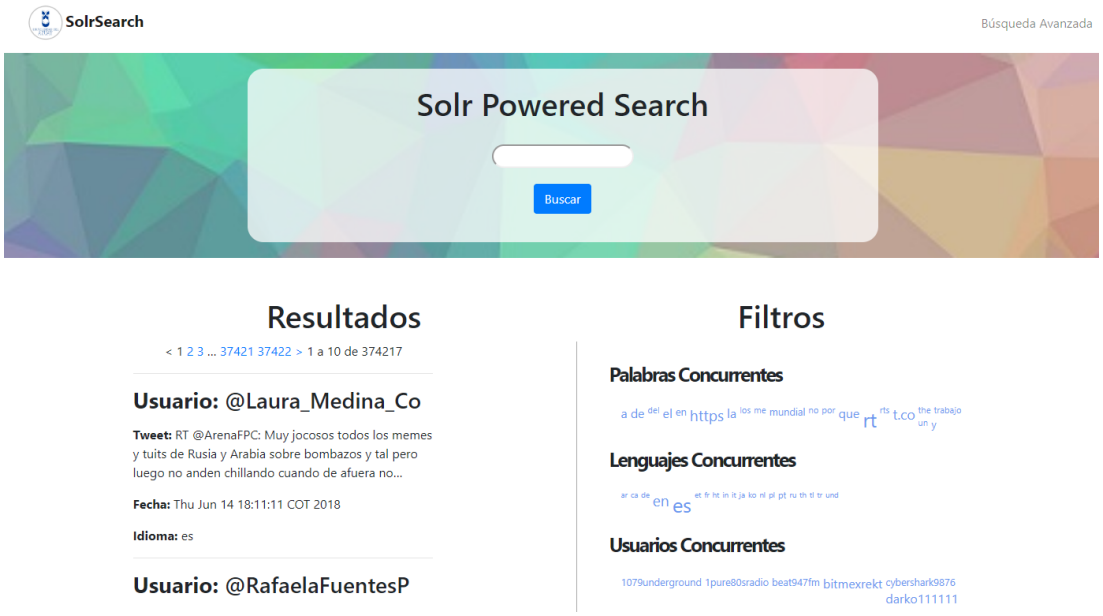


Ilustración 17 Recuperación de Información desde Cliente Web

4.2 Discusión

Los tweets recolectados en el sistema forman un set de datos en constante crecimiento, de los cuales muchos no cuentan con valor que justifique el consumo de recursos dentro de nuestra arquitectura, por esta razón autores como Hazra et al., (2015) realizan una búsqueda de tweets más específica para evitar recolectar datos que actúan como ruido. Estos autores hacen uso de usuarios para decidir qué datos tienen valor, eligiendo tweets denominados como tendencia dentro de la red social. Hacer esto también tiene sus contras debido a que se puede pasar por alto datos que tengan importancia social, política o económica los cuales sean menos populares dentro del segmento de la población que usa esta red social.

Los procesos de indexación y análisis son sensibles al contenido de los datos sobre los cuales se los efectúan. Es por esto que, para evitar ruidos autores como Disha et al., (2016), Sarnovsky et al., (2017) y Dudas et al., (2017), hacen uso de preprocesamiento y preparación de datos para evitar estos problemas. De esta forma se logra generar un set de datos más conciso y formalizado.

Como se ha manifestado a lo largo de este artículo los recursos necesarios para implementar un framework como Hadoop son muy altos, por esta razón sistemas como Kite para el manejo de datos de micro blogs Magdy, &Mokbel, (2017), optan por el uso de plataformas como Amazon Web Services (AWS) para montar sus arquitecturas distribuidas. Esto les brinda la posibilidad de escalar de manera muy fácil sin la necesidad de hacerse con recursos hardware, evitando así tareas como mantenimiento y configuraciones de equipos. La implementación de un Hadoop tanto en una arquitectura orientada a servicios o de manera tradicional, significan una inversión fuerte de dinero donde la decisión de implementación se deberá realizar en base al análisis de recursos y posibilidades de la organización que requiera implementarlo.

4.3 Conclusiones

En este artículo se ha propuesto una arquitectura basada en el framework de computación distribuida Hadoop y la plataforma de búsqueda Solr, para gestionar tweets generados por la red social Twitter. Esta arquitectura permite el manejo eficiente de Big Data proveniente de Twitter, gracias al uso de Hadoop sobre el cual realizan operaciones de obtención, indexación y almacenamiento de tweets en tiempo real, esto con el fin de conformar un set de datos políglota y multitemático aquel que facilite operaciones de recuperación de información para la búsqueda de tweets referentes a una temática en específico, permitiendo así la segmentación del set de datos en pequeñas colecciones de tweets que puedan ser usadas para la visualización o análisis de temas particulares. Concluimos que los datos provenientes de Twitter deben ser manejados necesariamente por una plataforma orientada al manejo de Big Data, esto debido a la cantidad de recursos que se requiere para su gestión, además de la necesidad de implementar índices para su óptima recuperación tomando en cuenta que los contenidos de estos tweets varían completamente dependiendo del usuario, haciendo difícil del uso del set de datos en su completitud sin ser segmentado antes. En futuros trabajos se puede hacer uso de procesos más sofisticados para el filtrado de tweets evitando así ruido innecesario en la plataforma, así como el uso de preprocesamiento de datos para disminuir el tamaño de datos a almacenarse y sobre todo mejorar los procesos de búsqueda de información al eliminar palabras como pronombres, signos y puntuaciones, otro tema a futuro es la utilización del set de datos para la aplicación de algoritmos y procesos de análisis para la conversión de esta información en potencial conocimiento.

ANEXOS

1. Experimentación

En la siguiente sección se incluirá información acerca de la implementación y configuraciones necesarias para las herramientas Hadoop, Solr, así como los componentes para extraer y consultar datos.

1.1. Implementación de Hadoop

La implementación de este *framework* es un proceso largo que involucra el invertir varias horas para su finalización, por este motivo se resaltan las fases importantes dentro de esta instalación.

1.1.1. Prerrequisitos de Instalación

Previa a la instalación de Hadoop necesitamos el preparar nuestros equipos con un sistema operativo Linux. El Hadoop montado para experimentar utilizó una distribución de Centos 7.

Todos los equipos involucrados en el clúster deben estar correctamente identificados con un nombre de host y una ip estática además de tener acceso a la Internet. En la ilustración 4 podemos observar el esquema de red usado para montar Hadoop.

Luego de configurar los equipos, se necesita obtener los archivos de Hadoop encontrados en el repositorio de Cloudera, esto se deberá hacerlo únicamente en el nodo identificado como Cloudera Manager. Para esto necesitamos ejecutar la siguiente sentencia en la terminal de comandos:

```
wgethttps://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo -P
/etc/yum.repos.d/
```

Una vez con el repositorio configurado podemos instalar el software de Cloudera Manager Server con el siguiente comando:

```
sudo yum installcloudera-manager-daemonscloudera-manager-server
```

Para una correcta funcionalidad del clúster necesitamos instalar el JDK de Oracle con este comando:

```
yuminstall oracle-j2sdk1.7
```

La distribución de Hadoop provista por Cloudera necesita de una base de datos externa la cuál manejará los metadatos del clúster. Para la experimentación se utilizó una base de datos MySQL que se instaló en el equipo host2.uda.com.

Para que el Cloudera Manager pueda hacer uso de la base de datos MySQL es necesario instalar su controlador jdbc.

Cloudera Manager no crea por defecto las bases de datos necesarias para su funcionamiento, por lo que es necesaria la creación manual de las siguientes tablas:

Tabla 3 Bases de Datos de Cloudera Manager

Servicio	Base de Datos	Usuario
Servidor Cloudera Manager	scm	scm
Monitor de Actividades	amon	amon
Gestor de Reportes	rman	rman
Hue	hue	hue
Servidor HiveMetastore	metastore	hive
Servidor Sentry	sentry	sentry
Servidor Navegador de Auditorías de Cloudera	nav	nav
Servidor Navegador de Metadatos	navms	navms
Oozie	oozie	oozie

Los nombres de base de datos así como sus usuarios son sugerencias propuestas en el manual de usuario de Cloudera, para más información visitar esta dirección: https://www.cloudera.com/documentation/enterprise/latest/topics/introduction.html.

Cada una de las bases de datos creadas deben contar con un usuario al cual se le concedan todos los permisos sobre ellas.

Como configuración final de MySQL se debe ejecutar el siguiente comando en cada una de las bases de datos creadas, eliminando las comillas y remplazando los textos con los correspondientes:

```
/usr/share/cmf/schema/scm_prepare_database.sh mysql 'Nombre de BD' 'Usuario de BD' 'Contraseña'
```

1.1.2. Instalación de Hadoop

Hasta este punto los prerrequisitos de la instalación están completos, lo siguiente es instalar Hadoop mediante el *wizard* que Cloudera Manager ofrece.

Para poder empezar con esta instalación lo primero que se debe hacer es ejecutar el siguiente comando para inicializar Cloudera Manager:

```
systemctlstartcloudera-scm-server
```

Se necesita ejecutar este comando para revisar el estado de Cloudera Manager:

```
systemctlstartcloudera-scm-server
```

Sabremos que este software se inicializó correctamente cuando en la terminal de comandos se presente esta línea: Started Jetty server.

Para poder acceder al gestor web de Cloudera Manager debemos tipear en un navegador la dirección ip del equipo en donde se lo instaló, añadido el puerto 7180, por ejemplo 10.10.220.51:7180.Las credenciales por defecto para hacer login son usuario admin, contraseña admin.

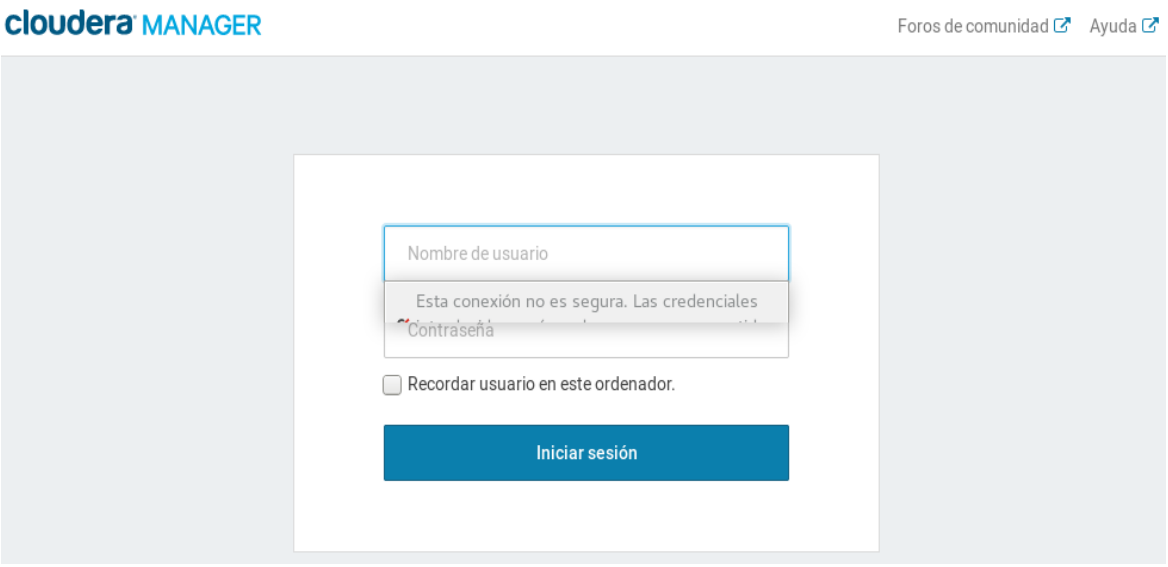


Ilustración 10 Formulario de inicio de sesión de Cloudera Manager

A partir de este punto la instalación es guiada a través de un wizard, por lo que solo se repasarán las pantallas importantes que necesitan de algún cambio, esto significa que otras pantallas encontradas en la instalación se pueden dejar con la configuración por defecto.

La siguiente pantalla requiere la especificación las direcciones ip o el nombre host de todos los nodos que formaran parte del clúster, exceptuando el equipo en donde se instaló Cloudera Manager.

Ilustración 11 Equipos que formarán parte del Clúster

Llegado a cierto punto Cloudera Manager empezará a descargar datos en cada uno de los hosts, por lo que esta pantalla dependerá de la velocidad de internet y del número de nodos que formen parte del clúster.

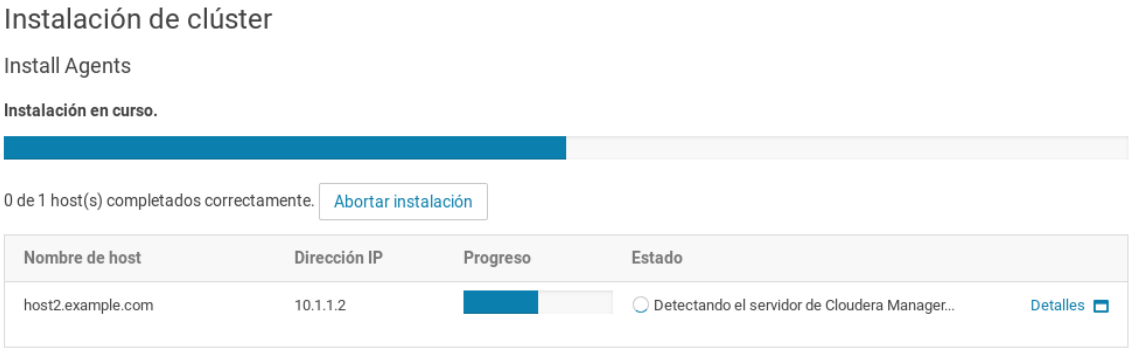


Ilustración 12 Instalación y configuración de Hadoop

Luego de varias pantallas de confirmación y servicios se presentarán cada uno de los roles disponibles de Hadoop.

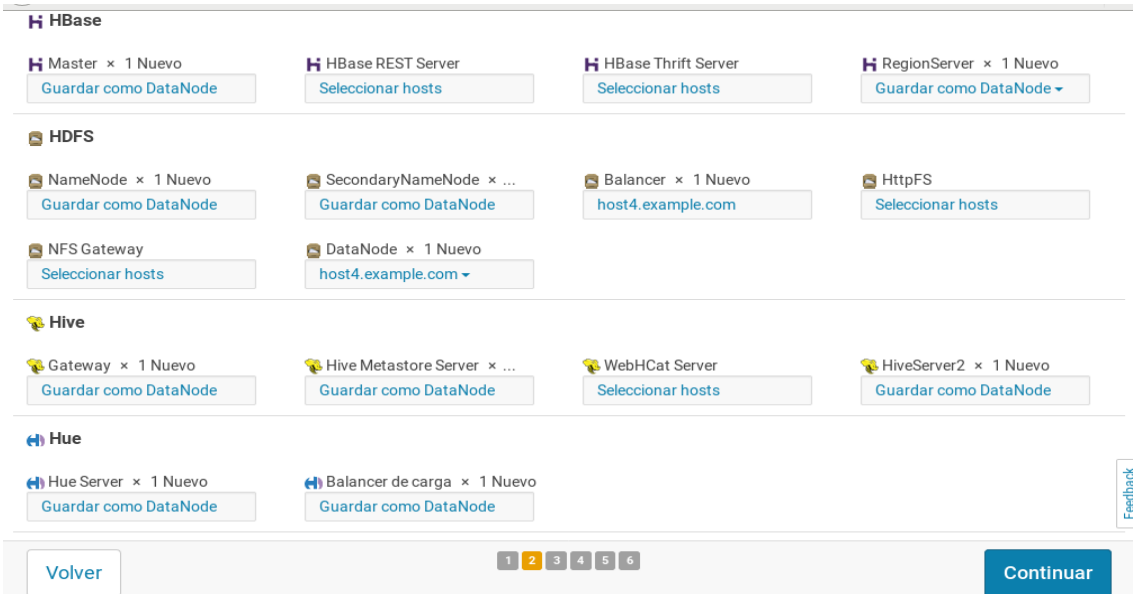


Ilustración 13 Selección de roles de Hadoop

Una vez en la siguiente pantalla se debe llenar los campos con el nombre de base de datos, usuario y contraseña configurados en pasos anteriores.

Ilustración 14 Configuración de base de datos

Por último, se realizará una ejecución de todos los servicios por primera vez, concluidos estos, Cloudera Manager, Hadoop, sus nodos y servicios estarán correctamente configurados y listos para usarse. La ejecución de los servicios suele fallar debido a falta de permisos en las carpetas creadas en cada uno de los nodos, esto se puede solucionar cambiando los permisos con el siguiente comando:

```
chmod 777 a /var/lib/´carpeta con el nombre del servicio´
```

1.2. Implementación del Indexador de Datos

Para crear un *core* de indexación lo primero que debemos hacer es generar archivos de configuración por defecto con el siguiente comando:

```
solrctlinstancedir --generate ´directorio´
```

En este comando se debe reemplazar la palabra entre comillas con cualquier directorio en nuestro equipo. Una vez terminado su ejecución tendremos varios archivos que necesitan modificarse para que Solr acepte el formato de nuestros datos.

En el directorio utilizado como referencia para la configuración tendremos un archivo XML con el nombre de *¨Shcema¨*, este es el encargado de definir que campos serán aceptados en los documentos a indexar, su tipo de dato y si son obligatorios o no. Para que los tweets puedan ser indexados por Solr es necesario el incluir todos sus campos dentro de este archivo de configuración, el formato con el cual se debe incluir cada uno de los valores es el siguiente:

```
<field name ="nombre_unico" type="string" indexed="true" required="true" stored="true"/>
```

Otro archivo de configuración a modificarse es el XML con el nombre de ¨solrconfig¨, este archivo indica a Solr donde debe almacenar los índices y sus datos. Este se debe estar configurado para utilizar HDFS como su almacén de datos por defecto.

Para cambiar esta configuración debemos identificar la sección de <directoryFactory> y reemplazarla con las siguientes líneas:

```
<directoryFactory name="DirectoryFactory"
        class="solr.HdfsDirectoryFactory">
        <str name="solr.hdfs.home">'dirección de Hdfs' </str>
        <bool name="solr.hdfs.blockcache.enabled">true</bool>
        <bool name="solr.hdfs.blockcache.read.enabled">true</bool>
        <bool name="solr.hdfs.blockcache.write.enabled">false</bool>
        <bool name="solr.hdfs.nrtcachingdirectory.enable">true</bool>
</directoryFactory>
```

En donde el valor de 'dirección de HDFS' debe referenciar en donde se almacenará los datos de Solr, por ejemplo: hdfs://10.10.120.53:8020/Solr /indexacion.

Con estas configuraciones listas podemos proceder a la creación del *core* de búsqueda. Para esto necesitamos asociar nuestra configuración a un core en ZooKeeper a través del siguiente comando:

```
solrctlinstancedir --create 'nombre_core' 'directorio'
```

Como último paso creamos el core en Solr, esto lo realizamos de la siguiente manera:

```
solrctl collection --create 'nombre_core' -s 1
```

Donde 'nombre_core' referencia como queremos llamar al *core* y la opción -s 1 refleja el número de instancias que tendrá este *core*, para la experimentación se utilizó solo una instancia de Solr.

A partir de aquí Solr es capaz de indexar tweets y almacenarlos en el HDFS de Hadoop.

1.3. Implementación del Extractor de Datos

Previamente al uso de la API y la descarga de información debemos obtener permiso para poder acceder a los datos de Twitter. Esto lo logramos mediante la creación de una aplicación en el sitio web de desarrolladores de Twitter que podemos encontrar en el siguiente enlace https://apps.twitter.com/.

La creación de la aplicación de desarrolladores pedirá el ingreso de los siguientes campos; 1) Name; 2) Description; 3) Website; 4) CallbackURL; en donde:

1) Name: Hace referencia al nombre de la aplicación.
2) Description: Una descripción del uso de la aplicación.
3) Website: Este campo indica el sitio web en el que se debe colocar la aplicación para su descarga, así como información de su uso y funcionamiento. Debido a que no se cuenta con un sitio web en este caso usamos una dirección ficticia.
4) Callback URL: En este campo se indica la URL a la cuál debe retornar luego de realizada la autentificación. Para la construcción del extractor de datos no es necesario el uso de una URL de retorno.

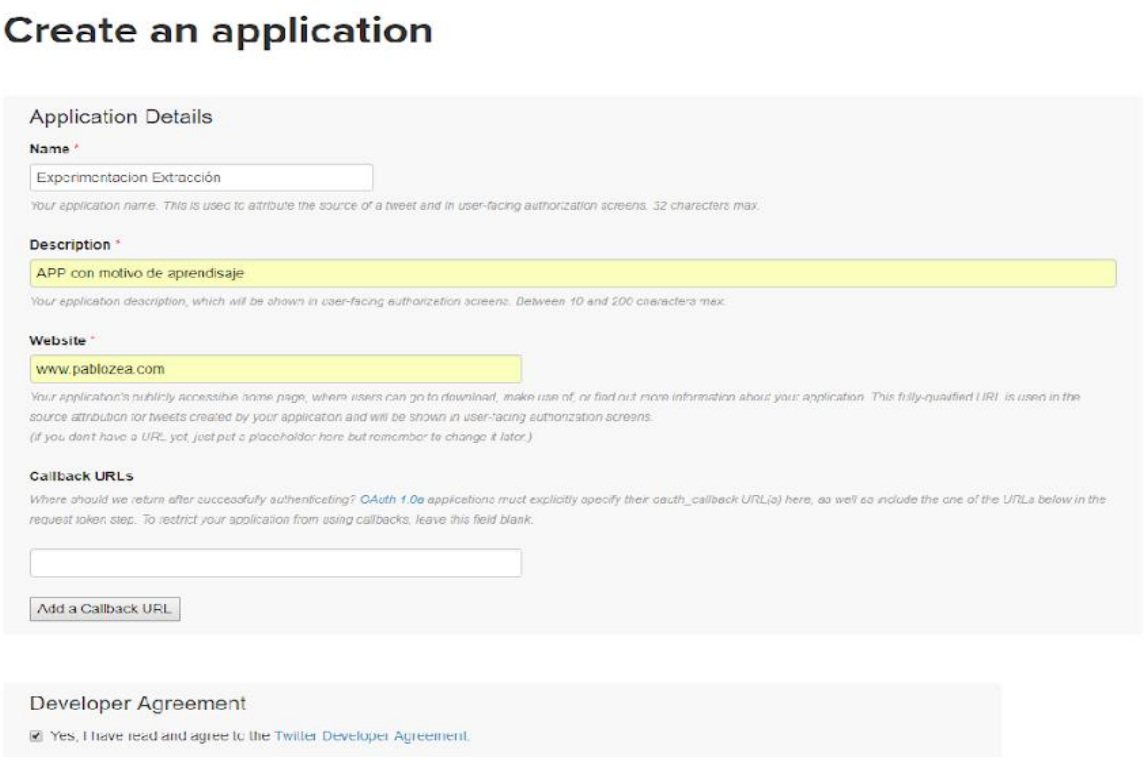


Ilustración 15 Creación de una aplicación de desarrolladores Twitter

Luego de creada la aplicación esta facilitará las credenciales para poder autentificar la API con nuestra aplicación de Twitter. Entre las credenciales obtenidas se encuentran; 1) Consumer Key; 2) ConsumerSecret; 3) Acces Token; 4) Access Secret.

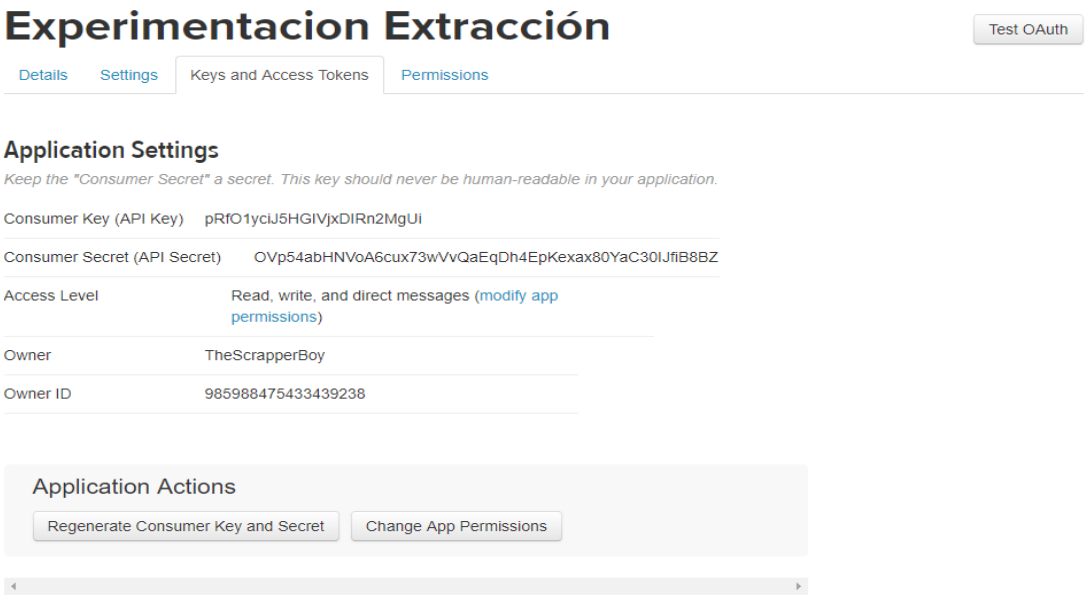


Ilustración 16 Credenciales de autentificación de la aplicación de Twitter

Para la construcción del extractor de datos utilizamos las siguientes sentencias:

1.3.1. Autentificación con Twitter

Este código realiza la autentificación de la API hacia la aplicación creada de Twitter, mediante el uso de las credenciales provistas por esta aplicación y la función de ConfigurationBuilder.

```
//Autentificación con la aplicación Twitter
ConfigurationBuildercb=new ConfigurationBuilder();
cb.setDebugEnabled(true)
.setOAuthConsumerKey("pRfO1yciJ5HGIVjxDIRn2MgUi")
.setOAuthConsumerSecret("OVp54abHNVoA6cux73wVvQaEqDh4EpKexax80YaC30IJfiB8BZ")
.setOAuthAccessToken("985988475433439238-OvOcCPDiYNY3ALs1mfpatzJ2LcV9S2B")
.setOAuthAccessTokenSecret("RlF4PIvNqU3pKXsPjyT2K8I4S8hfdhzQbu3l4IiH2GYvc");
```

1.3.2. Clientes Solr

La función HttpSolrClient crea un cliente Solr que permite la interacción hacia esta plataforma. Este cliente establece documentos con la función SolrInputDocument y los envía a la plataforma a que se indexen y almacenen con las funciones Solr.add y Solr.commit.

```
// Preparación de cliente Solr
StringurlString = "http://10.10.220.53:8020/solr";
SolrClientSolr = new HttpSolrClient.Builder(urlString).build();
// Preparación de documento de Solr
SolrInputDocumentdoc = new SolrInputDocument();
```

1.3.3. Extracción de datos

La extracción de datos en tiempo real se lo realiza mediante la función TwitterStreamFactory. Los campos de cada uno de los tweets se agregan a un documento del cliente Solr mediante la función doc.addField.

```
//Acceso a los datos de Twitter
TwitterStreamtwitterStream  = new TwitterStreamFactory(cb.build()).getInstance();
    StatusListener listener = new StatusListener() {
        public void onStatus(Status status) {
    //Añadir los datos del tweet que se enviará a indexarse y almacenarse
    doc.addField("id", status.getId());
            doc.addField("nom_usuario", status.getUser().getScreenName());
```

```
            doc.addField("text_tweet",status.getText());
            //Añadir en esta parte todos los campos que se deseen almacenar del tweet
        }
    //Se envían los tweets recuperados al Solr
    Solr.add(doc);
    Solr.commit();
```

1.3.4. Filtrado de tweets

La función FilterQuery permite el obtener tweets con temas referentes a ciertas palabras clave.

```
// Se filtra los tweets por medio de términos clave
FilterQueryfq = new FilterQuery();
String keywords[] ={"Ecuador,Cuenca,Mundial,Rusia,Tomebamba ,Comercio,Trabajo,@"};
fq.track(keywords);
twitterStream.addListener(listener);
twitterStream.filter(fq);
```

1.3.5. Compilación del proyecto

Para la creación del cliente Solr y la extracción de datos de Twitter se necesitó de la implementación de la dependencia 'Cliente solr' y 'Twitter4J'. A continuación, se muestra cómo se debe implementar la dependencia en el archivo Pom.xml.

1.3.5.1. Dependencia Solr

```
<dependency>
<groupId>Solr.client</groupId>
<artifactId>Solr.client</artifactId>
<version>1.6.6</version>
</dependency>
```

1.3.5.2. Dependencia Twitter4J

```
<dependency>
<groupId>Twitter4j </groupId>
<artifactId> Twitter4j </artifactId>
<version>4.2.5</version>
```

```
</dependency>
```

1.4. Implementación del Consultor de Datos

Para la implementación del consultor de datos se utilizó PHP, JavaScript y el Cliente AjaxSolr.

Este componente consta de los siguientes elementos:

1.4.1. Autentificación con Solr y Paginación de Datos

Este código realiza la autentificación hacia Solr y permite paginar todos los resultados devueltos por este:

```
$(function () {
  Manager = new AjaxSolr.Manager({
solrUrl: 'http://10.10.220.53:8983/solr/indexacion_tweets/'
  });
Manager.addWidget(new AjaxSolr.ResultWidget({
    id: 'result',
    target: '#docs'
  }));
Manager.addWidget(new AjaxSolr.PagerWidget({
    id: 'pager',
    target: '#pager',
prevLabel: '&lt;',
nextLabel: '&gt;',
innerWindow: 1,
renderHeader: function (perPage, offset, total) {
     $('#pager-header').html($('<span></span>').text( Math.min(total, offset + 1) + ' a ' +
Math.min(total, offset + perPage) + ' de ' + total));
}
  }));
```

1.4.2. Filtrado de resultados

El siguiente código permite el realizar filtros en base a las palabras más mencionadas dentro de los índices:

```
var fields = [ 'cont_tweet', 'usu_tweet', 'leng_tweet' ];
  for (var i = 0, l = fields.length; i< l; i++) {
Manager.addWidget(new AjaxSolr.TagcloudWidget({
    id: fields[i],
    target: '#' + fields[i],
    field: fields[i]
  }));
  }
Manager.init();
Manager.store.addByValue('q', '*:*');
  var params = {
   facet: true,
   'facet.field': [ 'cont_tweet', 'usu_tweet', 'leng_tweet' ],
   'facet.limit': 20,
   'facet.mincount': 1,
   'f.topics.facet.limit': 50,
'json.nl': 'map'
  };
```

1.4.3. Cliente Ajax Solr

Este cliente consiste de un conjunto de funcionalidades escritas en JavaScript las cuales se pueden encontrar en los siguientes archivos:

1) Reuters.1.js
2) ResultWidget.js
3) Manager.jquery.js
4) Core.js
5) AbastractManager.js

BIBLIOGRAFÍA

Al-Hajjar, D., &Syed, A. Z. (2015). Applying sentiment and emotion analysis on brand tweets for digital marketing. *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2015*. https://doi.org/10.1109/AEECT.2015.7360592

Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment. *Proceedings - 2016 IEEE 2nd International Conference on Big Data Computing Service and Applications, BigDataService 2016*, 52–57. https://doi.org/10.1109/BigDataService.2016.36

Jaywant, N., Shetty, S., &Musale, V. (2016). Digital identity based recommendation system using social media. *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (ICATccT)*, 288–292. https://doi.org/10.1109/ICATCCT.2016.7912010

Malas, M. D., &Vaidya, M. V. (2017). Real-Time Progressive Event Summarization and Sentiment Analysis on Evolutionary Tweet Stream, 388–393.

Murazza, M. R., &Nurwidyantoro, A. (2017). Cassandra and SQL database comparison for near real-time Twitter data warehouse. *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*, 195–200. https://doi.org/10.1109/ISITIA.2016.7828657

Dehdouh, K., Bentayeb, F., Boussaid, O., &Kabachi, N. (2014). Columnar NoSQL CUBE: Agregation operator for columnar NoSQL data warehouse. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *2014–Janua*(January), 3828–3833. https://doi.org/10.1109/SMC.2014.6974527

Lancieri, L., &Giovanetti, R. (2016). Multilevel exploration in Twitter social stream. *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, 726–731. https://doi.org/10.1109/ASONAM.2016.7752317

Barnaghi, P., Ghaffari, P., & Breslin, J. G. (2016). Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment. *Proceedings - 2016 IEEE 2nd International Conference on Big Data Computing Service and Applications, BigDataService 2016*, 52–57. https://doi.org/10.1109/BigDataService.2016.36

Dudas, P. M., Weirman, S., & Griffin, C. (2017). Little data, big stories: Taking the pulse of large-scaled events on twitter. *Proceedings - 2016 IEEE 2nd International Conference on Collaboration and Internet Computing, IEEE CIC 2016*, 474–482. https://doi.org/10.1109/CIC.2016.071

Disha, D. N., Sowmya, B. J., Chetan, &Seema, S. (2016). An efficient framework of data mining and its analytics on massive streams of big data repositories. *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 195–200. https://doi.org/10.1109/DISCOVER.2016.7806259

Desai, M., &Mehta, M. A. (2016). Techniques for sentiment analysis of Twitter data: A comprehensive survey. *2016 International Conference on Computing,*

Communication and Automation (ICCCA), 149–154. https://doi.org/10.1109/CCAA.2016.7813707

Lucas, G. M., Gratch, J., Malandrakis, N., Szablowski, E., Fessler, E., & Nichols, J. (2017). GOAALLL!: Using sentiment in the world cup to explore theories of emotion. *Image and Vision Computing*, *65*, 58–65. https://doi.org/10.1016/j.imavis.2017.01.006

Schmid, S., Galicz, E., & Reinhardt, W. (2015). WMS performance of selected SQL and NoSQL databases. *ICMT 2015 - International Conference on Military Technologies 2015*. https://doi.org/10.1109/MILTECHS.2015.7153736

Selvan, L. G. S., &Moh, T. S. (2015). A framework for fast-feedback opinion mining on Twitter data streams. *2015 International Conference on Collaboration Technologies and Systems, CTS 2015*, 314–318. https://doi.org/10.1109/CTS.2015.7210440

Greengrass, E. (2000). Information retrieval: A survey. *Information Retrieval*, (November), 1–224. https://doi.org/10.1.1.33.1855

Hendez, M., &Achour, H. (2014). Keywords extraction for automatic indexing of e-learning resources. *2014 World Symposium on Computer Applications and Research, WSCAR 2014*. https://doi.org/10.1109/WSCAR.2014.6916796

Qian, L., & Wang, L. (2010). An evaluation of Lucene for keywords search in large-scale short text storage. *2010 International Conference on Computer Design and Applications, ICCDA 2010*, *2*(Iccda), 206–209. https://doi.org/10.1109/ICCDA.2010.5541219

Gu, C., & Gao, Y. (2012). A Content-Based Image Retrieval System Based on Hadoop and Lucene. *2012 Second International Conference on Cloud and Green Computing*, 684–687. https://doi.org/10.1109/CGC.2012.33

Parveen, H., & Pandey, S. (2017). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. *Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology, ICATccT 2016*, 416–419. https://doi.org/10.1109/ICATCCT.2016.7912034

Patil, P., &Phursule, R. (2014). Survey Paper on Big Data Processing and Hadoop Components. *Ijsr.Net*, *3*(10), 585–590. Retrieved from http://www.ijsr.net/archive/v3i10/T0NUMTQyNTE=.pdf

Minanović, A., Gabelica, H., &Krstić, Ž. (2014). Big data and sentiment analysis using KNIME: Online reviews vs. social media. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, (May), 1464–1468. https://doi.org/10.1109/MIPRO.2014.6859797

Lai, W. K., Chen, Y. U., Wu, T. Y., &Obaidat, M. S. (2014). Towards a framework for large-scale multimedia data storage and processing on Hadoop platform. *Journal of Supercomputing*, *68*(1), 488–507. https://doi.org/10.1007/s11227-013-1050-4

Naik, N. (2017). Docker container-based big data processing system in multiple clouds for everyone. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*. https://doi.org/10.1109/SysEng.2017.8088294

Sarnovsky, M., Butka, P., &Huzvarova, A. (2017). Twitter data analysis and visualizations using the R language on top of the Hadoop platform. *SAMI 2017 - IEEE 15th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 327–332. https://doi.org/10.1109/SAMI.2017.7880327

Comparing Cloudera, Hortonworks, Amazon Web Services (AWS), Eastern Jin Technology Ltd. (Seabox Data), Exasol, Treasure Data, IBM, LexisNexis. (24 de Marzo de 2010). Recuperado de http://news.ninemsn.com.au/technology/10l31221/britain-launches-new-space-agency

Cunha, J., Silva, C., &Antunes, M. (2015). Health Twitter Big Bata Management with Hadoop Framework. *Procedia Computer Science*, *64*, 425–431. https://doi.org/10.1016/j.procs.2015.08.536

Azziki, A. E. L. F., Elabdallaoui, H. E., &Sadgal, M. (2016). Multi-Agent Frameworks in the service, 16–18.

Santos, C. Q., Tietzmann, R., Trasel, M., Moraes, S., Manssour, I. H., & Silveira, M. S. (2016). Can visualization techniques help journalists to deepen analysis of twitter data? exploring the «Germany 7 x 1 Brazil» case. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2016–March*, 1939–1948. https://doi.org/10.1109/HICSS.2016.245

Selvan, L. G. S., &Moh, T. S. (2015). A framework for fast-feedback opinion mining on Twitter data streams. *2015 International Conference on Collaboration Technologies and Systems, CTS 2015*, 314–318. https://doi.org/10.1109/CTS.2015.7210440

Hazra, T. K., Ghosh, A., Sengupta, A., &Mukherjee, N. (2015). Mitigating the adversities of social media through real time tweet extraction system. *2015 International Conference and Workshop on Computing and Communication, IEMCON 2015*. https://doi.org/10.1109/IEMCON.2015.7344483

Magdy, A., &Mokbel, M. F. (2017). Demonstration of kite: A scalable system for microblogs data management. *Proceedings - International Conference on Data Engineering*, 1383–1384. https://doi.org/10.1109/ICDE.2017.187

Doctora María Elena Ramírez Aguilar, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

C E R T I F I C A:

Que, el Consejo de Facultad en sesión del 11 de mayo de 2018, conoció y aprobó la solicitud para realización del trabajo de titulación, presentada por :

Estudiante: Pablo Andrés Zea Riofrío (cód. 66986)
Tema: **"Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información"**
Previo a la obtención del título de Ingeniero de Sistemas y Telemática
Director: Ing. Marcos Orellana
Tribunal: Ing. Gerardo Orellana
Ing. Pablo Pintado

Plazo de presentación del trabajo de titulación, con la calificación del Director: seis meses a partir de la fecha de aprobación, esto es hasta el 11 de noviembre de 2018.

E INFORMA:

Que en aplicación de la Disposición General Cuarta del Reglamento de Régimen Académico vigente, en caso de que el estudiante no culmine y apruebe el trabajo de titulación luego de dos períodos académicos contados a partir de la fecha de culminación de estudios, deberá realizar la actualización de conocimientos previa a su titulación.

Cuenca, 14 de mayo de 2018

UNIVERSIDAD DEL AZUAY | Ciencias de la Administración Facultad
Dra. María Elena Ramírez Aguilar
Secretaria - Abogada

Cuenca, 9 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

El tribunal para el trabajo de titulación denominado "Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información", presentado por Pablo Zea Riofrío, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, sugirió cambios en el diseño del trabajo de titulación. Debo ratificar que los cambios fueron realizados con los ajustes sugeridos, por lo que solicito se proceda con la aprobación por parte del Consejo de Facultad.

Atentamente,

Ing. Marcos Orellana Cordero
Director
Universidad del Azuay

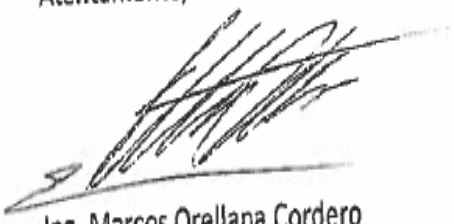
Cuenca, 4 de Mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De mis consideraciones:

Luego de revisar el diseño del trabajo de graduación denominado "Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información", presentado por Pablo Zea Riofrío, estudiante de la Escuela de Ingeniería de Sistemas y Telemática. Considero que el documento cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración, por lo que recomiendo su aprobación por parte del Consejo de Facultad.

Atentamente,



Ing. Marcos Orellana Cordero
Director

UNIVERSIDAD DEL AZUAY

DOCTORA MARÍA ELENA RAMÍREZ AGUILAR, SECRETARIA DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, el señor **ZEA RIOFRIO PABLO ANDRES** con código **66986,** alumno de la carrera de **INGENIERIA DE SISTEMAS Y TELEMATICA,** tiene aprobado el **91,11%** de créditos de su malla curricular.

Cuenca, 03 de mayo de 2018

Dra. María Elena Ramírez Aguilar
SECRETARIA DE LA FACULTAD
DE CIENCIAS DE LA ADMINISTRACIÓN

UNIVERSIDAD DEL AZUAY
FACULTAD DE ADMINISTRACIÓN
SECRETARÍA

Derecho No. **001-001-000171952**
mjmr.-

N° 0842228

UNIVERSIDAD DEL AZUAY

Cuenca 04 de Mayo del 2018

Ingeniero,

Oswaldo Merchán Manzano

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION

UNIVERSIDAD DEL AZUAY

De mi consideración,

Estimado Señor Decano, yo **Pablo Andrés Zea Riofrío** con C.I. **0105973523**, código estudiantil **66986**; estudiante de la carrera de Sistemas y Telemática; solicito muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema **"Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información"** previo a la obtención del título de Ingeniero en Sistemas y Telemática para lo cual adjunto la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipo mi agradecimiento.

Atentamente

Pablo Andrés Zea Riofrío

Estudiante de la Carrera de Sistemas y Telemática

0843473

UNIVERSIDAD DEL AZUAY

UNIVERSIDAD DEL AZUAY
INGENIERIA EN SISTEMAS Y TELEMATICA
DISEÑO DE TESIS

1. DATOS GENERALES

1.1 Nombre del estudiante: Pablo Andrés Zea Riofrío

1.1.1 Código: 66986

1.1.2 Contacto:

Teléfono convencional: 074096805

Celular: 0998770122

Correo electrónico: zea_pablo@yahoo.com

1.3 Director sugerido: Ing. Marcos Orellana Cordero

1.2.1 Contacto:

1.2.1.1 Correo: marore@uazuay.edu.ec

1.2.1.1 Teléfono: 0999955611

1.3 Co-director sugerido: Msc. Gerardo Orellana

1.3.1.1 Correo: gorellana@uazuay.edu.ec

1.3.1.2 Teléfono: 0998163786

1.4 Asesor metodológico:

1.5 Tribunal designado:

1.6 Aprobación:

1.7 Línea de Investigación de la carrera:

1.7.1 Código UNESCO:

1203 informática de computadores

1203.17 Informática

N° 0848369

UNIVERSIDAD DEL AZUAY

1.7.2 Tipo de trabajo: Investigación.

1.8 Área de estudio: Information Retrieval

1.9 Título propuesto: Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información.

1.10 Subtítulo:

1.11 Estado del proyecto: Previamente en la Universidad del Azuay se llevó a cabo el trabajo de grado Minería de texto en medios sociales. Caso de estudio del proyecto tranvía de Cuenca, mediante la utilización de la API provista por Twitter, se realizó la extracción de tweets sobre un tema polémico en la ciudad de Cuenca (Arias Belén, 2016), en la que debido a las limitaciones de las herramientas provistas por esta red social se acotó a aquellos publicados los últimos siete días a partir del momento en el cual se ejecuta el proceso de extracción. Esto puede conllevar a resultados erróneos al momento de analizar esta información, debido a que el proceso de extracción se puede ver sesgado por la falta de tweets previos, la fecha y circunstancias en el que estos hayan sido obtenidos. Se ve la necesidad de implementar un mecanismo que realice una constante verificación y extracción de nuevos tweets (Streaming). Dicha extracción de tweets aumentará de manera significativa el volumen de información almacenada, sin embargo, el encontrar y recuperar información de forma automática se vuelve en una tarea sin sentido si no se cuenta con un mecanismo de recuperación de información preciso. Por lo tanto, además de los métodos de streaming y almacenamiento es necesaria la aplicación de métodos de indexación de información para el filtrado y la búsqueda eficiente de tweets. Todo esto con el propósito de que la información obtenida sea lo más completa y objetiva posible, brindándonos los medios necesarios para la obtención de resultados lo más apegados a la realidad analizada.

Nº 0848356

UNIVERSIDAD DEL AZUAY

2. CONTENIDO

2.1 Motivación de la investigación: En la actualidad la cantidad de información disponible gracias a la internet es inmensa y en la mayoría de los casos es de índole público, dicho es el caso de la información contenida en red social Twitter, esta plataforma brinda acceso a un potencial conocimiento que día a día crece de manera desmesurada debido al gran número de usuarios y su constante actividad dentro de esta. Aunque mucha información disponible en la red social Twitter puede ser de ayuda para la generación de conocimiento, este suele encontrarse embebida entre información irrelevante como lo son anuncios u otra información proveniente de sitios web ajenos a esta fuente. La motivación de la investigación se enfoca en la extracción, indexación y estructuración de la información disponible en la red social Twitter, para la generación de un corpus de texto que en posteriores investigaciones se utilizará para la generación de conocimiento.

2.2 Problemática: La creciente información generada por la red social Twitter brinda la oportunidad de obtener conocimiento en tiempo real sobre una variedad de temáticas, sin embargo, la validez y calidad de este conocimiento es dependiente de la información obtenida a través de la API de esta red social, lo que produce que las limitaciones de extracción de tweets influyan directamente sobre los resultados de cualquier análisis realizado en estos datos, por no contar con todos los tweets necesarios como para no subestimar resultados inesperados. Por este motivo es necesario la utilización de un mecanismo de búsqueda y extracción de tweets en tiempo real también conocido como "Streaming", lo que permitirá dejar de lado el sesgo de información por falta de tweets sobre un tema en específico. Por otra parte el Streaming al estar constantemente extrayendo y almacenando tweets es acertado pensar que la cantidad de

información no estructurados se incrementara de una manera desmesurada, haciendo que los query de una base de datos no sean la herramienta adecuada para el filtrado y búsqueda de tweets compuestos de lenguaje natural, motivo por el cual se plantea el uso de métodos de indexación de documentos y la realización de un motor de búsqueda que será utilizado para filtrar tweets de una manera más precisa y efectiva.

2.3 Pregunta de investigación:

¿Cuáles son las técnicas o herramientas idóneas para la extracción continua de datos soportadas por la API de la red social Twitter?

¿Cuál es el tipo de bases de datos idóneo para el almacenamiento de grandes cantidades de tweets, siendo evaluadas bases de datos relacionales, no relacionales y RDF?

¿Cuál es la técnica de indexación de documentos más relevante dentro la comunidad científica y su pertinencia de uso en el caso propuesto?

UNIVERSIDAD DEL AZUAY

2.4 Resumen:

En la actualidad la red social Twitter es una fuente de información pública que está en constante crecimiento y actualización, estas cualidades hacen de esta red social el lugar óptimo para la obtención de información en tiempo real de una gran variedad de temáticas, sin embargo, debido a la vasta cantidad de datos no estructurados las tareas de identificación, extracción e integración se hacen complejas, lo que nos lleva a desaprovechar el potencial conocimiento contenido en Twitter, por ello se ve necesario el desarrollo de métodos automáticos de recuperación de información para la extracción de datos no estructurados además de la aplicación de técnicas de indexación para el desarrollo un motor de búsqueda el cual permita un filtrado eficiente y preciso de los tweets obtenidos a través de la API provista por Twitter. La información extraída de esta red social se incorporará a un corpus de texto que permitirá el filtrado de tweets de temáticas específicas, que en investigaciones posteriores se analizarán y procesarán para la generación de conocimiento.

2.5 Indagación Exploratoria:

Mucha de la información que se encuentra en redes sociales es abundante en temas de ocio o entretenimiento que poco pueden servir para la generación de conocimiento, sin embargo, debido a la penetración lograda en nuestra sociedad muchos medios y usuarios dependen de ellas para expresar sus opiniones sobre una vasta variedad de temas como lo pueden ser política, noticias, eventos, etc (Bolton, Parasuraman, Hoefnagels, Migchels, Kabadayi, Gruber, ... Solnet (2013)). Este comportamiento brinda la posibilidad de abastecerse de información pública y de libre uso la cual a través de técnicas adecuadas de extracción, indexación y análisis pueden revelar la situación actual sobre una temática específica.

La extracción de datos de una red social nos ofrece una colección de documentos textuales no estructurados que incrementan a un ritmo acelerado, motivo por el que se necesitan métodos

automáticos de recuperación de información (Information Retrieval), debido a que esta tiene como propósito la búsqueda de temas específicos en una colección dada que satisfaga la necesidad de información (Greengrass, 2000).

El campo de uso de la recuperación de información es bastante amplio siendo utilizado para la recolección de datos de varias de temáticas, como lo es el caso de la detección y búsqueda de noticias de última hora ocurridas en Twitter. Aquí se analizaron las características para proponer un método el cual sea capaz de coleccionar e indexar a aquellas noticias provenientes de esta red social (Phuvipadawat & Murata, 2010)., así también como el análisis de las opiniones de usuarios provenientes de Twitter frente a dos de las proveedoras más grandes de servicios en la nube, para lo cual se utilizó más de 1500 tweets dirigidos hacia cada proveedor con la finalidad de conocer que opinan y cómo se sienten los clientes de estas dos grandes proveedoras. (Qaisi & Aljarah, 2016).

Es evidente que las redes sociales como Twitter son una fuente inagotable para la aplicación de técnicas y herramientas de recuperación de información, sin embargo, debido a la naturaleza mayormente informal de Twitter en el cual usuarios se expresan sin limitaciones, hace posible la obtención datos para el enriquecimiento de un Corpus de texto para la realización de una minería de opiniones. (Pak & Paroubek, 2010).

La recuperación de información de redes sociales se puede realizar por medio de varias herramientas como web crawlers que con la ayuda del API de esta red social facilita la extracción de datos, siendo un limitante el número de peticiones e información que se puede acceder y extraer en un periodo continuo de tiempo (Hannon, Bennett & Smyth, 2010).

Al momento de realizar una recuperación de información de una red social como Twitter, uno de los mayores retos es el encontrar la manera de indexar esta información adecuadamente haciendo de esta visible en un grupo categorizado. Para la realización de esta tarea se pueden utilizar varios métodos como indexación a través de frases claves tópicas dentro de un tweet (Zhao, Jiang, He, Song, Achananuparp, Lim & Li, 2011), así como la indexación a través de una

organización de términos de diccionario. (Busch, Gade, Larson, Lok, Luckenbill & Lin, 2012), e incluso se hace uso de herramientas de indexación como lo es Lucene que proporciona acceso a potentes funciones de indexación y ponderación de términos (Hannon, Mike, Smyth, 2010).

2.6 Objetivo general:

- Implementar un procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información.

2.7 Objetivos específicos:

1. Indagar investigaciones sobre la generación e indexación de un corpus de texto mediante la red social Twitter.
2. Evaluar la idoneidad de los tipos de base de datos relacionales, no relacionales y RDF para la estructuración y almacenamiento de grandes cantidades de tweets.
3. Analizar tres de las principales técnicas de extracción de información en tiempo real aplicadas en la red social Twitter.
4. Analizar al menos dos técnicas de indexación de documentos con más relevancia dentro de la comunidad científica y su idoneidad sobre el caso de estudio propuesto.
5. Implementar un método de recuperación de información en tiempo real en la red social Twitter.
6. Implementar una técnica de indexación de documentos sobre la información obtenida de Twitter, además de un motor de búsqueda de tweets.

UNIVERSIDAD DEL AZUAY

2.8 Metodología:

Se partirá de una revisión bibliográfica de fuentes académicas, que permitan la identificación de las técnicas más relevantes de extracción y recuperación de información, mediante el uso de indexación de documentos.

Se experimentará sobre los métodos de extracción e indexación de datos considerados, determinando sus ventajas, limitaciones y su idoneidad de uso en la investigación propuesta.

Posteriormente se implementarán las técnicas seleccionadas para construir un sistema de extracción e indexación de tweets en tiempo real.

Se documentará y evaluará los resultados de las técnicas empleadas para la extracción e indexación de tweets dando como resultado un corpus de texto.

2.9 Alcances y resultados esperados:

Realizar un estado del arte que permita conocer que ha sido investigado en cuanto a la recuperación e indexación de Información de Twitter, con el objetivo de facilitar herramientas, métodos y conceptos necesarios para llevarlo a cabo.

Evaluar al menos tres métodos de recuperación de información de la red social Twitter, además de herramientas y lenguajes de programación para aplicarlos, con el fin de aplicar el método más eficiente en la extracción de tweets.

Evaluar al menos dos de los métodos más relevantes para la indexación de documentos, así como herramientas necesarias para aplicarlos, con la intención de poder llegar a filtrar de una manera precisa y eficiente los tweets almacenados.

Extracción, estructuración y almacenamiento de información de la red social Twitter en tiempo real.

Indexar los tweets obtenidos y almacenados de la red social Twitter, además de realizar un motor de búsqueda el cual permita filtrar y buscar tweets de una manera eficiente y precisa.

2.10 Supuestos y riesgos:

Riesgos	Probabilidad	Alternativas de solución
Riesgo de que la frecuencia de datos extraídos de la red social Twitter sea limitada debido a las políticas de estos sitios web.	Alta	Apegarse a las políticas de ratios de descarga de información de Twitter.
Riesgo de que las cuentas usadas para acceder a la información de la red social Twitter sean baneadas debido a la extracción de grandes cantidades de información.	Media	Implementar mecanismos que permitan el cesar la frecuencia de extracción de tweets.
Los métodos de indexación de documentos utilizados no sean los correctos y el filtrado de estos no sea del todo preciso.	Baja	Evaluar de manera exhaustiva las técnicas seleccionadas para la indexación de documentos.
La información extraída de Twitter abrume a la capacidad de procesamiento y almacenamiento con la que se cuenta actualmente.	Media	Seleccionar con detenimiento el lugar de almacenamiento de la información extraída.
La base datos y estructura utilizada para el almacenamiento de la información obtenida de Twitter no sea la adecuada, provocando cuellos de botella en el sistema.	Media	Evaluar y seleccionar con detenimiento la base de datos sobre la cual se almacenarán los tweets.

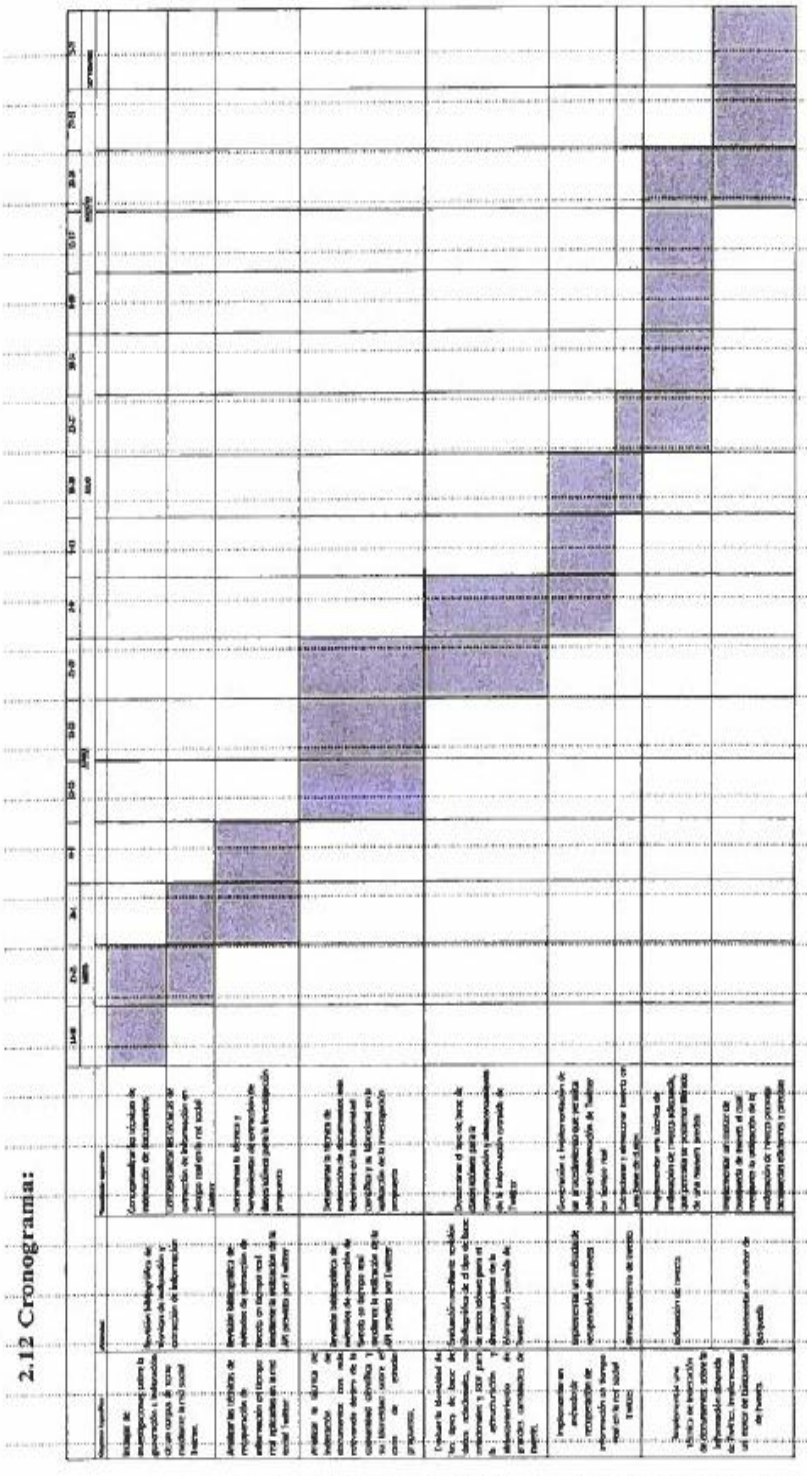
UNIVERSIDAD DEL AZUAY

2.11 Esquema tentativo:

- **Capítulo 1:** Introducción
- **Capítulo 2:** Estado del Arte
- **Capítulo 3:** Extracción, estructuración y almacenamiento de tweets
- **Capítulo 4:** Recuperación e indexación de la información obtenida a través de la red social Twitter.
- **Capítulo 5:** Análisis de resultados.
- **Conclusiones**
- **Bibliografía**

UNIVERSIDAD DEL AZUAY

2.12 Cronograma:



UNIVERSIDAD DEL AZUAY

2.13 Referencias:

Greengrass, E. (2000). Information retrieval: A survey. *Information Retrieval*, (November), 1–224. https://doi.org/10.1.1.33.1855.

Phuvipadawat, S., & Murata, T. (2010). Breaking news detection and tracking in Twitter. *Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2010*, 120–123. https://doi.org/10.1109/WI-IAT.2010.205.

Xin, W., Jing, Z., Jing, J., Yang, H., & Palakorn, S. (2011). Topical Keyphrase Extraction from Twitter, 379–388.

Busch, M., Gade, K., Larson, B., Lok, P., Luckenbill, S., & Lin, J. (2012). Earlybird: Real-time search at Twitter. *Proceedings - International Conference on Data Engineering*, 1360–1369. https://doi.org/10.1109/ICDE.2012.149.

Hannon, J., Bennett, M., & Smyth, B. (2010). Recommending twitter users to follow using content and collaborative filtering approaches. *Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys '10*, 199. https://doi.org/10.1145/1864708.1864746

Rieder, B. (2013). Studying Facebook via data extraction. *Proceedings of the 5th Annual ACM Web Science Conference on - WebSci '13*, 346–355. https://doi.org/10.1145/2464464.2464475.

Qaisi, L. M., & Aljarah, I. (2016). A Twitter Sentiment Analysis for Cloud Providers: A Case Study of Azure vs. A WS, 1–6.

Pak, A., & Paroubek, P. (n.d.). Twitter as a Corpus for Sentiment Analysis and Opinion Mining, 1320–1326.

Doan, A., Naughton, J. F., Ramakrishnan, R., Baid, A., Chai, X., Chen, F., … Vuong, B. Q. (2009). Information extraction challenges in managing unstructured data. *SIGMOD Rec.*, *37*(4), 14–20. https://doi.org/10.1145/1519103.1519106

Bolton, R. N., Parasuraman, A., Hoefnagels, A., Migchels, N., Kabadayi, S., Gruber, T., … Solnet, D. (2013). Understanding Generation Y and their use of social media: a review and

research agenda. *Journal of Service Management*, *24*(3), 245–267. https://doi.org/10.1108/09564231311326987

Arias, Z. M. (2016). Minería de texto en medios sociales. Caso de estudio del proyecto tranvía de Cuenca.

UNIVERSIDAD DEL AZUAY

2.14 Anexos: para casos en los que se requiera respaldar el proyecto.

2.15 Firma de responsabilidad (estudiante)

Pablo Andrés Zea Riofrío

2.16 Firma de responsabilidad (director sugerido)

Ing. Marcos Orellana Cordero

2.17 Firma de responsabilidad (Asesor Metodológico)

2.18 Fecha de entrega:

09/05/2018

N 0843474

CONVOCATORIA

Por disposición de la Junta Académica de la escuela de Ingeniería de Sistemas y Telemática se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: "**Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información**", presentado por el estudiante Pablo Andrés Zea Riofrío con código 66986, previa a la obtención del título de Ingeniero de Sistemas y Telemática, para el día **Miércoles, 09 de mayo de 2018 a las 08:20.**

Tomar en cuenta que posterior a la sustentación del Diseño del Trabajo de Titulación, por ningún concepto se puede realizar modificaciones ni cambios en los documentos; únicamente, en caso de diseño aprobado con modificación, el Director adjuntará al esquema un oficio indicando que se procede con los cambios sugeridos.

Cuenca, 07 de mayo de 2018

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

Ing. Marcos Orellana .....

Ing. Gerardo Orellana .....

Ing. Pablo Pintado .....

FECHA: 07 de MAYO DE 2018.

Estudiante: Pablo Zea Riofrío

Oficio Nro. 021-2018-DIST-UDA

Cuenca, 3 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 3 de mayo del 2018, recibió el proyecto de tesis titulado "Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información", presentado por Pablo Zea Riofrío, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Marcos Orellana, codirector y miembro de tribunal al Ing. Gerardo Orellana, y como miembro de tribunal al Ing. Pablo Pintado.

Atentamente,

Ing. Catalina Astudillo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

Ing. Esteban Crespo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

Fecha de sustentación: **Miércoles, 09 de mayo de 2018 a las 08:20**

1.40. Nombre del estudiante: Pablo Andrés Zea Riofrío

1.41. Código: 66986

1.42. Director sugerido: Ing. Marcos Orellana

1.43. Codirector (opcional): _____

1.43.1. Tribunal: Ing. Gerardo Orellana e Ing. Pablo Pintado

1.43.2. Título propuesto: **"Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información"**

1.43.3. Aceptado sin modificaciones :

_____

1.43.4. Aceptado con las siguientes modificaciones:

OBJETIVO ESPECIFICO 4: BUSQUEDA DE NUEVOS DOS FORMAS DE INDEXACION DE DOCUMENTOS CON MAS RELEVANCIA DENTRO DE LAS CONSULTAS CIENTIFICAS Y SU IDONEIDAD SOBRE EL CASO DE ESTUDIO PROPUESTO.

1.43.5. No aceptado

1.43.6. Justificación:

_____

_____

_____

Tribunal

Ing. Marcos Orellana

Ing. Gerardo Orellana

Ing. Pablo Pintado

Sr. Pablo Zea Riofrío

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN
(Tribunal)

1.1. Nombre del estudiante: Pablo Andrés Zea Riofrío
1.2. Código : 66986
1.3. Director sugerido:
1.3.1. Codirector (opcional): Ing. Marcos Orellana
1.4. Título propuesto: **"Procedimiento de descarga e indexación de tweets mediante métodos de recuperación de la información"**
1.5. Revisores tribunal: Ing. Gerardo Orellana e Ing. Pablo Pintado
1.6. Recomendaciones generales de la revisión:

	Cumple	No cumple
Problemática y/o pregunta de investigación		
1. ¿Presenta una descripción precisa y clara?	✓	
2. ¿Tiene relevancia profesional y social?	✓	
Objetivo general		
3. ¿Concuerda con el problema formulado?	✓	
4. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓	
Objetivos específicos		
5. ¿Permiten cumplir con el objetivo general?	✓	
6. ¿Son comprobables cualitativa o cuantitativamente?	✓	
Metodología		
7. ¿Se encuentran disponibles los datos y materiales mencionados?	✓	
8. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓	
9. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓	
10. ¿Las técnicas planteadas están de acuerdo con el tipo de investigación?	✓	
Resultados esperados		
11. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓	
12. ¿Concuerdan con los objetivos específicos?	✓	
13. ¿Se detalla la forma de presentación de los resultados?	✓	
14. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionados?	✓	

Ing. Marcos Orellana Ing. Gerardo Orellana Ing. Pablo Pintado