UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

Implementación de métodos, herramientas y técnicas para el análisis de *Big Data*

TESIS
que para optar por el grado de Maestría en Ciencia e Ingeniería de la Computación

Presenta:
Juan Garfias Vázquez

Tutor:
Dra. María del Pilar Angeles Facultad de ingeniería

Ciudad Universitaria. Cd. Mx., Enero 2018

Agradecimientos

A la Universidad Nacional Autónoma de México, que a lo largo de mi vida ha sido un segundo hogar.

A mis amigos y compañeros, a quienes el destino me ha presentado. Gracias por compartir conmigo consejos, reuniones y pláticas que han aligerado las clases, tareas y exámenes. Gracias por su amistad.

Por sobre todo gracias a mis padres, Lidia Vázquez Sanchez y Juan Garfias Zarate, y a mi hermano Eder Garfias Vázquez. Gracias por su infinito amor, por su ejemplo de entrega y dedicación a sus labores, por la guía, la ayuda y el apoyo constante que me han dado durante toda mi vida. Gracias por inculcarme el amor a la verdad en sus distintas formas. Esta tesis es también suya.

Indice de contenido

Indice de ilustraciones

Indice de tablas

Introducción

A través de la historia, el desarrollo tecnológico ha tenido una evolución muy importante en temas sobre el análisis de los datos. Han surgido áreas de estudio como la minería de datos, la inteligencia de negocios, el *Data Warehouse*, ciencia de datos, *Big Data*, etc.

La presente tesis se enfocará en el estudio de datos masivos, mejor conocida como *Big Data*, implementando un caso práctico que servirá para evaluar las herramientas, explorar las técnicas y realizar un análisis sobre datos masivos o *Big Data*.

El capítulo 1, contiene el marco teórico, que respalda la implementación y el desarrollo con el que fue realizado el caso práctico. Se explican los conceptos más importantes que definen a un proyecto de *Big Data*, y los ciclos de vida para el desarrollo de un proyecto de ese tipo. Se explica el análisis de sentimiento y su importancia en su aplicación. Se describe a *Hadoop*, como una de las tecnologías desarrolladas más importante para el análisis de datos masivos, ya que un gran número de herramientas se conectan con este contenedor de datos. Se aborda el tema de *MapReduce*, el cual hace posible el procesamiento masivo y en paralelo de los datos. Se explica cómo funciona *Hive* y su lenguaje *HiveQL* para realizar consulta en los contenedores de datos de *Hadoop*. Se describe el marco de trabajo que contiene un gran número de herramientas configuradas para el procesamiento de datos masivos llamado *Hortonworks*.

El capítulo 2, contiene antecedentes de estudios realizados sobre datos masivos, e implementado en redes sociales, específicamente en twitter. Se explican los objetivos de cada una de las investigaciones, cómo se implementaron y qué resultados se obtuvieron de los análisis de datos realizados.

El capítulo 3, documenta el inicio del ciclo de vida del proyecto de *Big Data*, en donde se realiza el análisis del proyecto que será implementado, se plantean los objetivos y una serie de preguntas que deberá de contestar el proyecto, y se definen los alcances. Se diseña las partes de los componentes, como son los diagramas de bases de datos, los diagramas de secuencia, y por último, se propone la arquitectura que se implementará para el desarrollo del proyecto.

El capítulo 4, documenta el desarrollo de las aplicaciones diseñadas en el capítulo anterior, explicando el uso del *API* de *twitter*, el analizador de sentimiento utilizado y la integración de los programas en su conjunto, para conectarse con el marco de trabajo de *Hortonworks*. Por último, se da una breve explicación del uso de *Hive*, *Pig Latin* y *Zeppelin*.

El capítulo 5 y último, documenta el análisis realizado a los datos, ejecutando las consultas al contenedor de *Hadoop* para contestar las preguntas planteadas en el capítulo 3, y se muestran los resultados de manera gráfica.

Capítulo 1. *Big Data*

El presente capítulo presenta las bases necesarias para abordar el problema que va a ser planteado y resuelto con las técnicas, herramientas y metodologías para el análisis de datos masivos mejor conocida como *Big Data*. Como objetivo principal de esta tésis, es documentar cómo se puede realizar una implementación de un proyecto de esta naturaleza, investigando antecedentes de estudios similares y ver qué resultados generaron, para así poder diseñar y modelar una solución genérica que pueda ser adaptable o ampliada para generar más conocimiento en base a los resultados obtenidos.

Otro objetivo, es conocer el funcionamiento técnico y teórico de las herramientas a ser utilizadas para el análisis, las cuales son un tema bastante extenso y complejo, ya que cada herramienta tiene antecedentes científicos y técnicos, por lo que se abordarán a un grado que ayude a la comprensión de su rol en la implementación.

El análisis de datos masivos requiere de una base de conocimientos técnicos y teóricos, y sobre todo si se quiere abordar temas complejos en áreas de conocimiento, en donde el insumo son los datos para realizar análisis en ellos mismos, como son las áreas de minería de datos, ciencia de datos o inteligencia de negocios. Por lo tanto, la justificación de la tesis es plantear una base de la cual partiendo de un proyecto de análisis de *Big Data*, este pueda convertirse en un proyecto de cualquier otra área que tenga como fin generar conocimiento a partir de grandes cantidades de datos.

Las redes sociales son un gran generador de datos, por lo que para esta tésis, se utilizará como proveedor de datos a Twitter, ya que esta red social nos ofrece una conexión de flujo constante de datos a las publicaciones que realizan sus usuarios en tiempo real. En el desarrollo de la tesis, se va a realizar un análisis de los datos obtenidos con herramientas libres que se mencionarán más adelante en este capítulo.

De manera general, la forma de trabajo del proyecto de *Big Data* que esta tesis va a tratar, es primero que nada, una investigación de las herramientas existentes y sus antecedentes en cuanto a quien las desarrolló, cómo las desarrollaron y que problemáticas impulsaron su desarrollo. Posterior a conocer las herramientas y sus antecedentes, se debe hacer una investigación sobre metodologías documentadas para que conceptualmente se plantee el flujo de trabajo que se va a seguir, para tener ubicadas las etapas del desarrollo de un proyecto de esta naturaleza. Los antecedentes sobre trabajos similares son importantes, por lo que se realiza una investigación sobre proyectos similares y documentados, para tomarlos como base y realizar el diseño y sobre todo, un despliegue de resultados satisfactorio.

La contribución de esta tésis es dar un panorama de estudios sobre *Big Data*, mostrar cómo interpretar las etapas de una metodología definida, implementar técnicas de análisis de datos y utilizar herramientas desarrolladas para el procesamiento de grandes volúmenes. Todo esto con el fin de proponer una estructura replicable y de fácil entendimiento para emprender un proyecto similar sobre el análisis de datos en Twitter.

Entonces, para estar en contexto en el área de estudios de bases de datos, se ha visto que ha crecido el volumen de datos de manera acelerada, y adopta la ley de Moore, formulada por el cofundador

de Intel Gordon E. Moore, en la cual expresa que aproximadamente cada dos años, el número de transistores en un microprocesador se duplica (Moore, 1964).

Adaptando al contexto de la ley de Moore a los datos en general, generados y almacenados, estos crecen de manera similar o superior. Si comparamos el desarrollo de las tecnologías de almacenamiento del año 2017 con el 2006, las memorias flash eran de entre 16 a 32 megabytes y comparándolo con los discos duros de mayor uso de la época en computadoras personales, eran de una capacidad aproximada de 40 a 60 gigabytes, por lo que si comparamos con las capacidades actuales en donde tenemos memorias flash de hasta 2 terabytes y discos duros de hasta 8 terabytes, se puede comprender que hay una gran capacidad para almacenar una inmensa cantidad de archivos de todo tipo, con contenidos variados y en formatos innumerables.

Si consideramos la utilización de teléfonos móviles con capacidades inteligentes, la conectividad a redes de internet inalámbricas y las redes sociales, nos podemos dar cuenta que producimos una gran cantidad de datos diariamente. Por lo tanto, así como consumimos agua, generamos basura o gastamos dinero, ahora puede sumarse a nuestros consumos los datos que generamos al día.

En el 2010, la revista *Good & Oliver Munday*, en colaboración con IBM, publicó el monto masivo de generación de datos por los seres humanos, exponiendo las siguientes cifras:

- 2.9 millones de correos electrónicos enviados cada segundo.
- 375 megabytes consumidos por los hogares.
- 20 horas de video en YouTube por cada minuto.
- 24 peta bytes de datos procesados por google.
- 50 millones de tuits por día.
- 700 mil millones de minutos se pasan en Facebook los usuarios.
- 1.3 exabytes se han enviado y recibido por medio de teléfonos inteligentes.
- 72.9 productos son ordenados en Amazon por segundo (A collaboration between GOOD and Oliver Munday, in collaboration with IBM., 2010).

Debido a que se están generando datos de todo tipo y de diferentes formas, el procesamiento, análisis y obtención de un valor agregado o extracción de conocimiento se convierten en grandes retos. A raíz de la necesidad de procesar y analizar inmensos conjuntos de datos en tiempo real para la toma de decisiones, han desarrollado métodos y herramientas para el análisis de *Big Data*.

1.1 ¿Qué es *Big Data*?

El término *Big Data* se basó en el artículo de Doug Laney, quien realizó una recopilación de los principales problemas que tenían las empresas para la administración de sus datos, las cuales se enfrentaban a lo que denominó como *3D* de administración de datos: dicho término se refiere al volumen, la velocidad y la variedad de los datos. Es decir, para que se pueda aplicar el término de *Big Data*, el procesamiento de grandes volúmenes de información debe ser en tiempo real y la información debe ser variada en tipo de dato (Laney, 2001).

Danah Boyd y Kate Crawford, mencionan que el *Big Data* no se distingue por el tamaño sino por la capacidad de ser relacionada con otros datos. Debido a los esfuerzos sólo por minar y agregar datos, el *Big Data* está muy relacionado con servicios en la red. Sus valores provienen de patrones, que

pueden ser derivados de hacer conexiones con otros conjuntos de datos sobre individuos, individuos en relación con otros, grupos de personas o simplemente sobre la estructura de la información misma. Por otro lado, definen *Big Data* como algo cultural, tecnológico y un fenómeno académico que se basa en la interacción de tipo:

- Tecnológica: Maximizar el poder computacional y la precisión de los algoritmos para reunir, analizar, relacionar y comparar grandes conjuntos de datos.
- Analítica: Identificando patrones basándose en grandes conjuntos de datos para hacer afirmaciones en lo económico, social, técnico y legal.
- Mitología: La mayoría cree que un conjunto grande de datos ofrece una forma elevada de inteligencia y conocimiento, que puede generar ideas que son previamente imposibles con sensación de verdad, objetividad y precisión (Boyd & Kate, 2011).

En el comercio electrónico, como afirma Min Chen, es un buen ejemplo para entender cómo de una transacción se pueden obtener datos para almacenarlos y estudiarlos posteriormente, esto hace entender a una empresa que los datos mismos ya son un activo tangible, y se empeñan en conservarlos (Min Chen, 2014).

Almacenar los datos, implica pagar un espacio de almacenamiento de acuerdo a lo que se requiere, pero esto conlleva a que entre más crezca el volumen de los datos, entonces el valor de cada fuente de datos se reduce, ya que de un contenedor de datos, la mayor parte son datos adicionales necesarios para los procesos, pero innecesarios para el análisis, por lo que los datos innecesarios crecen más rápido que los datos útiles, resultando en una mala justificación financiera el incrementar el almacenamiento en línea.

Para mitigar el crecimiento acelerado de los contenedores de datos, se sugiere que se implementen procesos que limiten el recolectado de datos, analizando qué realmente tiene valor, eliminando redundancias, monitoreando el uso de los datos y determinar cuáles están sin utilizar para ser eliminados.

La característica de velocidad, por otro lado, independientemente de la red de banda ancha o la arquitectura implementada, va más relacionada con la capacidad de administrar la velocidad de respuesta. Esto quiere decir que para que sea aceptable se recomienda que los datos estratégicos se extraigan y se integren en otro contenedor de datos, como por ejemplo utilizar memoria caché para el acceso instantáneo de datos, sin degradar el desempeño de los contenedores de datos, o los enlaces punto a punto entre contenedores de bases de datos que se utilizan para realizar análisis estratégicos, y diseñar una arquitectura balanceada entre la aplicación y los requerimientos de los datos, sin asumir que la información entera deba estar al alcance en tiempo real.

Y por último, la variedad de los datos, la cual no debe de ser barrera para una administración efectiva, a pesar de existir formatos de datos incompatibles, datos no estructurados e inconsistencia en la semántica, por lo que se crean mecanismos para relacionarlos y resolver las inconsistencias.

Entonces, al haber conocido las características que un modelo de *Big Data* debe tener, podemos entender que no sólo se trata de conjuntos grandes de datos, sino que a ese conjunto de datos sin estudiar ni procesar, se convertirá en un modelo de *Big Data* hasta que enfrentemos el reto de almacenar, procesar y estudiar los grandes conjuntos de manera eficiente, por lo que recientemente

Min Chen propuso (2014) *4V*, es decir, Volumen de datos, Variedad de tipos de datos, Velocidad de respuesta y Valor agregado que sea de utilidad para la toma de decisiones (Min Chen, 2014).

Min Chen, hace referencia a lo que se puede lograr al aprovechar los métodos de *Big Data*. En las operaciones de los gobiernos europeos se podrían ahorrar más de 100 billones de euros, reduciendo fraudes, errores y diferencias en los impuestos. Hasta este momento, la multimedia, redes sociales, el creciente auge del internet de las cosas, están haciendo que las empresas coleccionen más información, en un crecimiento exponencial en su volumen, por lo que el *Big Data* crecerá su potencial en crear valor para las empresas y los consumidores (Min Chen, 2014).

En el artículo que publicó Jingran Li, menciona tres aspectos importantes, que ha resumido como lo que está existiendo en las aplicaciones de *Big Data*, al seguir las fases de la administración del ciclo de vida del producto (PLM). El primer aspecto se refiere a la administración de los datos y a la programación de tareas sobre el *Big Data*, menciona que en los procesos de construcción de las aplicaciones, se deben modelar contenedores de datos que puedan trabajar con más datos, más velocidad y más usuarios, y por otra parte las tareas programadas, deben facilitar el dinamismo y las características del tiempo real. El segundo aspecto menciona la Administración de la Cadena de Suplementos (SCM), el cual es un concepto de logística, pero basado en el contexto de *Big Data*, el cual hace el énfasis en que las redes de las compañías deben de colaborar juntas. Y por último, el tercer aspecto es la aplicación de *Big Data* para la personalización masiva (MC), el cual consiste en entregar productos a los clientes, con la capacidad de ser personalizados a través de buenos procesos de integración y flexibilidad (Li, Fei, Ying, & Liangjin, 2014).

La Administración del Ciclo de vida del Producto, tiene como objetivo lograr una rápida inserción en el mercado, alta calidad, bajo costo, el mejor servicio, el ambiente más limpio, gran flexibilidad y alta generación de conocimiento.

Jingran Li menciona, que la quinta *V* sobre las características del *Big Data*, la define como la *variabilidad*, que se refiere a la expansión en el rango de valores posibles de los datos, por lo que pueden cubrir el rango completo de la experiencia humana, o en otras palabras, el número de valores posibles puede ser infinitamente amplio. *Big Data* siempre mostrará más varianza que los conjuntos de datos tradicionales (Li, Fei, Ying, & Liangjin, 2014).

1.2 Impacto en la sociedad

El Foro Económico Mundial hace mención del impacto del *Big Data* en una amplia variedad de giros, como los servicios financieros, educación, salud, agricultura etc., los cuales están aprovechando el uso de los teléfonos inteligentes que han aumentado considerablemente su uso entre la población, ya que para las personas con menos recursos es el único medio de interacción con la tecnología, lo que al mismo tiempo está permitiendo recopilar una inmensa cantidad de datos de los usuarios, de tal manera que al analizarlos se pueden detectar comportamientos para así determinar las necesidades de la población. La privacidad y seguridad de los datos es un factor importante, ya que se debe definir qué datos se utilizarán para realizar análisis en beneficio de la sociedad, los cuales no formarán parte ya que también se está haciendo un énfasis en la privacidad de los usuarios. Por otro lado, compañías de redes sociales, ofrecen sus servicios sin ningún costo para el usuario final, ya que para otros actores, los datos recabados son muy valiosos (WEForum, 2012).

1.3 Fundamentos de *Big Data*

En el libro de Thomas Erl menciona que la adopción del *Big Data* además de poder transformar, tiene principalmente la capacidad de innovar el negocio dado que requiere un cambio de mentalidad, también implica alterar la estructura, sus productos, servicios y la organización misma. (Erl, Buhler, & Khattak, 2016)

El ciclo de vida que plantea Erl inicia estableciendo el caso de negocio de *Big Data*, y termina con asegurar que los resultados son dados a la organización para que puedan producir con su máximo valor. Antes de realizar el análisis de los datos, se requiere un número de etapas compuesta por los pasos de identificación, obtención, filtrado, extracción, limpieza y agregación de datos, que se detallarán más adelante.

La organización como requisito para implementar un proyecto de *Big Data* debe tener una administración de datos y un marco de trabajo para *Big Data*. Los datos no actualizados, inválidos o mal identificados en la entrada, producirán resultados de baja calidad, de igual manera, el tiempo de vida del ambiente de *Big Data* debe ser planificado, así como definir un plan de trabajo para asegurar que cualquier expansión necesaria pueda ser sincronizada con los requerimientos de la organización.

La adquisición de los datos mismos puede ser económica, debido a la disponibilidad de plataformas libres, herramientas de código abierto y el uso de hardware económico. Sin embargo, se debe considerar que si se quiere dar mayor valor a los datos obtenidos, se deben obtener datos de terceros. El hecho de tener un gran volumen y variedad de datos puede ayudarnos a tener mayor posibilidad de encontrar señales ocultas en los patrones que se detecten.

La privacidad es un factor importante a considerar, ya que las bases de datos pueden contener información confidencial, o también al ser relacionadas con otras bases de datos, pueden extraer patrones de comportamiento. Por ejemplo, si se tiene una base de datos de traslados de un automóvil mediante su GPS, y ésta se relaciona con el dueño y sus compras realizadas con tarjetas bancarias, podríamos determinar cuándo viaja, a qué establecimientos va, qué compra en determinados días de la semana, etc., lo cual para la persona sería una invasión a su privacidad. En cuanto al tema de seguridad, se deben utilizar los protocolos de autenticación para acceder a los datos, definir responsables de los mismos y asegurar el medio de comunicación, sobre todo cuando se utilicen las redes de internet.

La procedencia de los datos se refiere a que debemos tener información de nuestra fuente de datos y cómo los procesa. Conocer la procedencia de los datos nos ayudará a determinar la autenticidad y calidad de los mismos. En las diferentes etapas en el ciclo de vida del análisis, los datos se pueden encontrar en diferentes estados: trasmitido o *data-in-motion*, en procesamiento o *data-in-use* y en almacenamiento o *data-at-rest.*

Existe la necesidad de tener resultados en tiempo real o lo más cercano al tiempo real, para mandar alertas en las aplicaciones o en los paneles de control. La mayoría de las soluciones de *Big Data* de código abierto están orientadas a cargas por lotes, aunque hay una nueva generación de herramientas de código abierto con capacidades de tiempo real para procesar flujos de datos y analizarlos en el instante.

El desempeño rápido se vuelve un reto dado el volumen de datos que se requiera procesar. Por ejemplo, realizar el procesamiento de datos con algoritmos complejos implicará un largo tiempo de espera para obtener el resultado, otro ejemplo sería la capacidad de transferencia de la red, ya que al tener grandes volúmenes de datos, estos deben de ser transferidos de un contenedor a otro, lo cual implica una demora considerable, ya que si se quisiera transferir 1 petabyte en una red con capacidad de 1 Gbps, nos resulta en 2,750 horas aproximadamente para transferir ese volumen de datos.

Un marco de trabajo de gobierno es requerido, esto para asegurar que los datos, y el ambiente de la solución desarrollado, pueda ser regulado, estandarizado y evolucionado de una manera controlada.

Por otra parte, se debe definir la metodología para controlar cómo los flujos de datos van a entrar y salir de la solución de *Big Data*, cómo será la retroalimentación entre el personal de TI y los usuarios de la aplicación para mejorar e incrementar el valor de los resultados.

El uso de la nube provee ambientes remotos que pueden contener infraestructura de TI, almacenamiento y procesamiento a gran escala, una gran alternativa para ser utilizada por una organización en un proyecto de *Big Data*, ya que las principales justificaciones para su uso son: tener hardware inadecuado, capital insuficiente para la infraestructura, los contenedores de datos igual están en la nube, las capacidades de almacenamiento y cómputo disponibles tienen un límite bastante alto.

1.3.1 Ciclo de vida según Min Chen

En el 2014 Min Chen define el ciclo de vida de un proceso de *Big Data* en las siguientes etapas:

1. Generación de datos: Este primer paso consiste en la manera en la que se generan los datos, ya que actualmente, las fuentes son muy amplias a partir de que se empezaron a desarrollar nuevas tecnologías para la automatización, monitoreo, comunicaciones, etc.
2. Adquisición de los datos: Este paso se implementa a la hora de recibir los datos, a los que se le aplican diversos procesos para almacenar únicamente lo que es útil, para que no impacte en el desempeño del análisis que se efectúe. Se divide en tres métodos:
    a. Colección de los Datos: Es la fuente de donde serán obtenidos como popularmente son los archivos de tipo *log* en texto plano, o el formato en el que se encuentra definida una estructura como los archivos separados por espacios o caracteres especiales.
    b. Trasmisión de los datos: También conocido como transporte, es el proceso en el que los datos son filtrados y recopilados de diversas fuentes en un solo contenedor para ser analizados. El transporte de los datos puede ser por medio de redes o medios de almacenamiento externos.
    c. Pre-procesamiento de los datos: Este método es uno de los más importantes, ya que, el resultado de esta actividad impacta directamente a los resultados del análisis que se vaya a efectuar. Los procesos de extracción, transformación y carga, por sus siglas en ingles *ETL*, son los que predominan en la integración de datos. Se busca limpiar los datos dependiendo de la fuente, ya que en ciertos casos pueden contener errores al ser ingresados por personas, provocando de esta manera redundancia, duplicidad e inconsistencia de los datos, por lo que esta tarea puede

ser muy compleja y se deben diseñar, en algunas ocasiones algoritmos especializados para tareas particulares de limpieza.

3. Análisis de los datos: Este paso tiene una gran variedad de posibilidades, ya que existen diferentes métodos para procesar los datos que ya estaban previamente almacenados por una integración, existen métodos dependiendo de la necesidad, como puede ser que se quieran obtener resultados de análisis en tiempo real, análisis fuera de línea, análisis a nivel de memoria, análisis de inteligencia de negocios, y análisis masivos utilizando archivos HDFS cuando la escala de los datos es ampliamente rebasada para los productos de inteligencia de negocios, o por los sistemas manejadores de bases de datos tradicionales (Min Chen, 2014).

1.3.2 Ciclo de vida según Thomas Erl

En el libro de Thomas Erl *Big Data Fundamentals: Concepts, Drivers & Techniques*, nos menciona que el análisis de datos tradicional es muy diferente al que se realiza sobre *Big Data*, principalmente porque este último se diferencia en el volumen, la velocidad y la variedad de los datos. Para realizar un análisis de datos masivos, se requiere una metodología que guíe las tareas y las actividades relacionadas con la adquisición, procesamiento, análisis y reutilización de datos (Erl, Buhler, & Khattak, 2016).

El ciclo de vida que propone Thomas Erl se divide en nueve etapas, como se muestra en la Ilustración 1, que se describe a continuación:

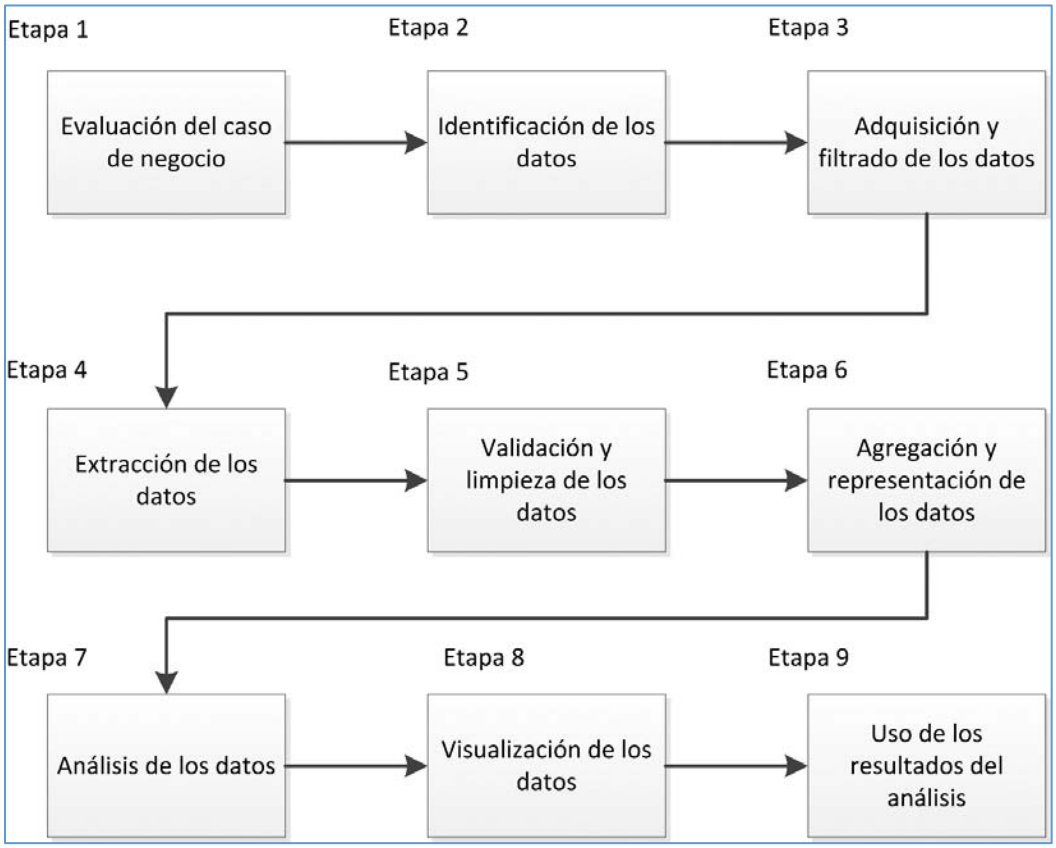


Ilustración 1. Las nueve etapas del ciclo de vida de Thomas Erl. (2016)

1. Evaluación del caso de negocio:
    a. En esta etapa inicial se debe definir el caso de negocio y debe estar claramente entendida la necesidad del análisis.
    b. La evaluación del caso de negocio ayudará a los que toman decisiones en la organización, a entender cuáles serán los recursos necesarios a utilizar y las metas que serán atacadas por el análisis.
    c. La capacidad de entender los objetivos del proyecto en esta etapa, pueden ayudar a determinar el criterio y el enfoque en el cual se basará el análisis.
    d. De igual manera, en esta etapa se debe identificar si el problema a resolver cae dentro de una o varias de las características básicas que definen a un proyecto de *Big Data*, las cuales, como se mencionó anteriormente, son volumen, velocidad y variedad de los datos.
    e. Otro punto importante a considerar en esta etapa es la inversión que esté dispuesta la organización a realizar en hardware, herramientas de software, capacitación, etc, y que esta retornen beneficios de las metas alcanzadas.
    f. Por último, es importante en el inicio de esta etapa realizar una investigación de las tecnologías de *Big Data*, como por ejemplo saber qué productos están disponibles, y qué capacitación será necesaria a lo largo de las etapas para impulsar el proyecto.
2. Identificación de los datos:
    a. En esta etapa se buscan identificar los datos necesarios, y de qué fuentes serán obtenidas para el proyecto de análisis y dependiendo de los objetivos del proyecto, se evaluará si se requieren fuentes de datos internas o externas a la organización.
    b. En caso de utilizar fuentes internas de datos, como pueden ser datos respaldados y datos de operación, estos se relacionan, reorganizan y concentran bajo especificaciones predefinidas.
    c. En caso de utilizar fuentes externas de datos, se enlistan los posibles proveedores de datos, como pueden ser concentradoras de datos públicos como blog y redes sociales, y estas se filtrarán y pre-procesarán con herramientas automatizadas para su adquisición.
    d. Mientras más amplias sean las fuentes de datos, incrementaremos la probabilidad de encontrar patrones ocultos y relaciones.

3. Adquisición y filtrado de los datos:
    a. En esta etapa, los datos son reunidos de todas las fuentes identificadas anteriormente. Los datos adquiridos son enviados a un filtro automatizado para renovar todos aquellos datos corruptos o que se identifiquen sin valor para los objetivos del análisis.
    b. Dependiendo de las fuentes de datos, estas pueden venir en archivos cuando son compradas por proveedores externos o en otros casos requieren una integración de una *API* para consultarla, como en el caso de la *API* de twitter.
    c. Hay que considerar que algunos datos innecesarios para nuestro análisis pueden ser necesarios para otros, por lo que se recomienda conservar una copia de los datos originales previo a realizar el filtrado de los mismos.

    d. Para los análisis por lotes de archivos, estos datos son almacenados en disco y procesados posteriormente. A diferencia de los casos de análisis en tiempo real, los datos son analizados primero y después almacenados en disco.
    e. Los metadatos, tanto para las fuentes internas o externas, pueden ser añadidos de forma automática para mejorar la clasificación y consulta de los datos.
    f. Es de vital importancia que los metadatos puedan ser leídos mecánicamente y pasados a lo largo de las etapas del análisis. Esto ayudará a mantener la procedencia de los datos a través del ciclo de vida, para establecer y preservar la certidumbre y calidad de los datos.
4. Extracción de los datos:
    a. En algunas ocasiones, los datos de entrada pueden llegar en formatos incompatibles con la solución de *Big Data*, por lo que a esta etapa se dedica a extraer de los diferentes tipos de datos y transformarlos en un formato que la solución de *Big Data* pueda utilizar para realizar el análisis.
5. Validación y limpieza de los datos:
    a. Datos inválidos pueden sesgar y falsear los resultados del análisis. A diferencia de los datos de una organización, los cuales tienen una estructura predefinida y son validados previamente. Sin embargo, la entrada de datos de un análisis de *Big Data* pueden ser no estructurados y sin alguna indicación de validez, por lo que su complejidad puede hacer difícil el llegar a un conjunto apropiado de restricciones y de validaciones.
    b. Por lo tanto, esta etapa está dedicada a establecer a menudo reglas de validaciones complejas y de remover datos inválidos conocidos.
    c. Las soluciones de *Big Data* a menudo reciben datos de diferentes fuentes. La redundancia entre las fuentes puede ser explotada explorando los conjuntos interconectados, con el objetivo de crear parámetros válidos y llenar los datos faltantes con datos válidos.
    d. Datos inválidos pueden ser reflejados en resultados incongruentes al momento de realizar el análisis.
6. Agregación y representación de los datos.
    a. Los datos pueden estar propagados en múltiples fuentes de datos, lo cual requiere relacionarlos por medio de campos comunes entre las fuentes, por ejemplo un identificador único o una clave única.
    b. En esta etapa se realiza la integración de múltiples conjuntos de datos para llegar a una vista unificada de los mismos.
    c. Realizar esta etapa puede ser complicada porque se pueden presentar diferencias en:
        i. Estructura de datos: A pesar de que el formato de los datos sea el mismo, el modelo de datos puede ser diferente.
        ii. Semántica: Un valor que es nombrado de diferente manera en dos conjuntos de datos y significar lo mismo, por ejemplo *surname* y *lastname*, que para ambos casos es el apellido de una persona.
    d. El procesamiento de datos en las soluciones de *Big Data* pueden hacer la agregación de datos una operación de mucho esfuerzo y tiempo. Relacionar las diferencias

puede requerir complejidad lógica que es ejecutada automáticamente sin la necesidad de la intervención humana.

7. Análisis de los datos:
    a. Esta etapa se dedicará a realizar las tareas de análisis de los datos, las cuales típicamente involucran uno o más tipos de análisis.
    b. Pueden ser en iteraciones, especialmente si el análisis de datos es exploratorio, esto quiere decir que se realiza un análisis repetitivo hasta que se encuentre un patrón de correlación satisfactorio.
    c. Dependiendo del tipo de resultados analíticos requeridos, esta etapa puede ser basada en consultas de agregación, realizando comparaciones en los conjuntos de datos. Por otro lado, puede ser combinado con minería de datos y análisis estadísticos complejos con el objetivo de encontrar patrones y anomalías, o también generar modelos matemáticos o estadísticos para representar relaciones entre las variables.
    d. El análisis de datos puede ser clasificado como análisis confirmatorio o análisis exploratorio:
        i. Análisis confirmatorio: Es un enfoque deductivo en donde la causa del fenómeno que está siendo investigado es propuesta previamente. La causa propuesta es llamada hipótesis. Los datos son entonces analizados para aprobar o desaprobar la hipótesis y de proveer respuestas definitivas a preguntas específicas. Las técnicas de muestreo de datos son típicamente usadas. Hallazgos inesperados o anomalías son ignoradas desde que la causa predeterminada fue asumida.
        ii. Análisis exploratorio: Es un enfoque inductivo que está relacionado cercanamente con la minería de datos. No hay hipótesis predefinidas, por lo que los datos son explorados a través del análisis, desarrollando un entendimiento de la causa de un fenómeno. Por lo tanto, esto podría no proveer respuestas definitivas, este método provee una dirección general de lo que puede facilitar el descubrimiento de patrones o anomalías.
8. Visualización de los datos:
    a. La habilidad de analizar datos masivos y encontrar señales tendrá poco valor, si la única persona que puede interpretar los resultados es el analista de los datos.
    b. Por lo tanto, esta etapa está dedicada a utilizar las técnicas de visualización de datos y herramientas para comunicar gráficamente los resultados de manera efectiva, para los usuarios responsables del negocio.
    c. El usuario de negocio necesita ser capaz de entender los resultados para darles valor y posteriormente pueda tener la habilidad de proveer una retroalimentación.
    d. Los mismos resultados pueden ser representados de diferentes maneras, por lo que es importante usar la más adecuada técnica de visualización para mantener el dominio del negocio en el mismo contexto.
    e. Otro aspecto para tener en mente es que se provea un método para indagar en los datos originales para compararlos con simples estadísticas, con el objetivo de que los usuarios entiendan cómo y de dónde se obtuvieron los resultados.
9. Uso de los resultados del análisis:

a. Los resultados del análisis permite a los usuarios del negocio, tomar decisiones soportadas por los datos, y buscar nuevas oportunidades de utilización de dichos resultados.
b. En esta etapa, se dedicará a determinar cómo y dónde el resultado de los datos puede tener un mayor impacto.
c. Se debe identificar la forma en que los resultados impactaran en los procesos de negocio.

1.4 *Big Data* en la administración del ciclo de vida del producto

La Administración del Ciclo de Vida del Producto (PLM, por sus siglas en inglés) tiene su origen a principios del siglo veintiuno para administrar el proceso intensivo de conocimiento, que consiste principalmente en el análisis del mercado, el diseño del producto y el proceso de desarrollo, la manufactura del producto, la distribución del producto, el producto en uso, el servicio post-venta y el reciclaje del producto. Es decir, administra los productos a través de los ciclos de vida.

El ciclo de vida de un PLM inicia por el diseño de una idea, la cual posteriormente se detalla la descripción de la misma para después pasar a la fase de producción. El producto es almacenado en un contenedor de productos para después ser transportado al cliente en la fase de logística. En la fase de utilización, el cliente utiliza el producto mientras se le provee servicio remoto por los productores. Si algo sale mal entonces se procede a dar mantenimiento y si no se puede utilizar más, entonces procede a cerrar su ciclo de vida para ser reciclado o eliminado (Li, Fei, Ying, & Liangjin, 2014).

Las fases descritas anteriormente se dividen dentro de tres periodos:

- Inicio de la Vida (BOL, por sus siglas en inglés): BOL es el periodo en que el concepto del producto es generado, diseñado y subsecuentemente físicamente realizado.
- Mitad de la Vida (MOL, por sus siglas en inglés): MOL es el periodo en el que los productos son distribuidos, usados y mantenidos por los clientes o los ingenieros.
- Final de la Vida (EOL, por sus siglas en inglés): EOL es cuando el producto es reciclado por los productores o desechados por los clientes.

1.4.1 Los datos en BOL, MOL y EOL

Para lograr un buen desempeño en el PLM hay que averiguar qué tipo de datos están involucrados antes de proponer métodos analíticos avanzados. Esto ayuda para que las técnicas de *Big Data* resuelvan los problemas relacionados con el producto o puedan tomar cierta decisión basada en los tremendos contenedores de datos en diferentes niveles.

A. BOL, inicio de la vida

En el inicio del ciclo de vida recae en dos pasos esenciales que son el análisis del mercado o también conocido como el dominio del problema, en donde lo más importante es conocer las demandas del cliente que vienen de diferentes tipos. La variedad de los datos incluyen comentarios en blogs, videos subidos en internet, marcas en los sitios web o las relaciones en los comportamientos de las ventas. Además, si en el análisis del mercado hay demandas faltantes entonces la información para el MOL y EOL presentará deficiencias, lo cual se reflejará en el desempeño del producto.

En el paso del diseño del producto, los datos involucrados pueden ser definidos desde la descripción de las necesidades a la descripción de la función del producto y por último al diseño detallado de las especificaciones usando diagramas sobre la configuración del producto, definir la manera correcta de codificar para el equipo de desarrollo y todo tipo de parámetros (Li, Fei, Ying, & Liangjin, 2014). En la Tabla 1 se muestran varios tipos de datos de entrada y salida del BOL.

Tabla 1. Datos en BOL. (Li, Fei, Ying, & Liangjin, 2014)

Datos de Entrada	
Categoría	**Datos principales**
Demandas del cliente	Funciones del producto, configuración de los paquetes, calidad, costo, marca y otros tipos relacionados.
Mantenimiento e información de fallas	Desglose de los problemas principales, frecuencia de mantenimiento, evaluación de fallas, lista de componentes críticos principales, origen de las causas, etc.
Información de colaboración corporativa	Información de proveedores y de *outsourcing* alternativos.
Datos de Salida	
Categoría	**Datos principales**
Especificación del Diseño	Lista de materiales, lista de proveedores, diagramas, códigos de programación, parámetros de configuración, parámetros de localización, parámetros de tolerancia, intensidad de los materiales, etc.
Información de la Producción	Instrucciones de ensamble, especificaciones de producción, datos de la historia de la producción, plan de producción, estatus del inventario, etc.

B. MOL, mitad de la vida

En la mitad del ciclo de vida el producto ya existe en su forma final, los problemas concernientes al servicio se han vuelto más significantes y necesitan tener mayor atención.

En la fase de logística, se procede a tomar decisiones y plantear estrategias para optimizar el contenedor de los datos y el transporte de los mismos, ya que desde la tendencia de la globalización de la logística, los datos han incrementado extremadamente el volumen. Los datos de entrada de esta fase son ordenados y lo que requieren los desarrolladores son arreglos óptimos. El cómo transformar información ordenada en arreglos inteligentes con la vista global es la tarea más crucial en esta fase.

En la fase de utilización, basándose en la información del manual del usuario, el cliente puede operar el producto con normalidad. En el proceso, el estatus de la información del producto será generado

para retroalimentar al desarrollador, quien hará lo posible para involucrarse en la fase de utilización y en dado caso de ser necesario, dar soporte generando los manuales necesarios para crear modelos de mantenimiento (Li, Fei, Ying, & Liangjin, 2014). En la Tabla 2 se muestran varios tipos de datos de entrada y salida del MOL.

Tabla 2. Datos en MOL. (Li, Fei, Ying, & Liangjin, 2014)

Datos de entrada	
Categoría	**Datos principales**
Manual del usuario	Introducción de las funciones del producto, guía de instalación, condiciones de uso, precauciones, etc.
Información de producción	Instrucciones de ensamble, especificaciones de producción, datos históricos de la producción, plan de la producción, estatus del inventario, etc.
Información del soporte y mantenimiento	Lista de repuestos, precios de las partes de repuesto, instrucciones de mantenimiento y servicio, etc.
Datos de salida	
Categoría	**Datos principales**
Información del estatus del producto	Grado de calidad de cada componente, definición de desempeño, etc.
Información del ambiente del uso	Condiciones de uso (Por ejemplo el promedio de la humedad, temperatura externa e interna), perfil de la misión del usuario, tiempo de uso, etc.
Plan de mantenimiento	Ingenieros de mantenimiento, herramientas, fechas, lugares, costos, causas de las fallas, etc.

C. EOL, fin de la vida

Cuando el producto entra en la etapa final del ciclo de vida, las decisiones se concentran en lo que se recicla o se desecha. El estatus de degradación de cada componente es calculado desde el periodo del MOL. Con el propósito de maximizar los valores de los productos del EOL, se deben considerar las opciones adecuadas para las opciones de recuperación, reciclaje, reúso, re-manufactura, y desecho.

Ayudándonos de métodos analíticos, una óptima selección se basa en ¿cuándo, cómo, dónde y qué? del reciclaje puede ser obtenido (Li, Fei, Ying, & Liangjin, 2014). En la Tabla 3 se muestran varios tipos de datos de entrada y salida del EOL.

Tabla 3. Datos en EOL. (Li, Fei, Ying, & Liangjin, 2014)

Datos de entrada

Categoría	Datos principales
Información de la historia del mantenimiento	Identificadores de los componentes en problemas, fechas de instalación, identificadores de los ingenieros de mantenimiento, lista de los componentes remplazados, estadísticas de degradación después de la sustitución, costo del mantenimiento, etc.
Información del estatus del producto	Grado de calidad de cada componente, definición de performance, etc.
Información del ambiente de uso	Condiciones de uso (Por ejemplo el promedio de la humedad, temperatura externa e interna), perfil de la misión del usuario, tiempo de uso, etc.
Datos de Salida	
Categoría	**Datos principales**
Información de las partes recicladas	Reúso de una parte o componente, información de re fabricación, calidad de la re fabricación de una parte o componente.
Información del estatus del producto del fin del ciclo de vida	Tiempo de vida del producto, parte o componente, grado de reciclaje y reúso de cada parte o componente, etc.
Información del desmantelamiento	Facilitar la desmantelación, valor del reúso o reciclamiento, costo del desmantelamiento, costo de re fabricación, costo del desecho, etc.

1.4.2 Framework de *Big Data* en PLM

En el periodo del BOL, las principales fases son el diseño y la producción. El análisis del mercado y diseño del producto conforman la fase de diseño, mientras que la fase de producción implica la adquisición, fabricación de productos y la gestión de equipos.

En el período de MOL, que consiste en la logística, utilización, y las fases de mantenimiento, el *Big Data* presenta un enorme potencial en la gestión de almacenes, transporte del producto, entrenamiento del producto, soporte del producto, y el mantenimiento predictivo y preventivo.

En el período de EOL, cuando el único enfoque es cómo procesar los productos obsoletos, el *Big Data* juega un papel importante en la toma de decisiones de recuperación del producto EOL y el plan de logística inversa. (Li, Fei, Ying, & Liangjin, 2014)

Todas estas actividades y sus aplicaciones se pueden apreciar en la Ilustración 2.

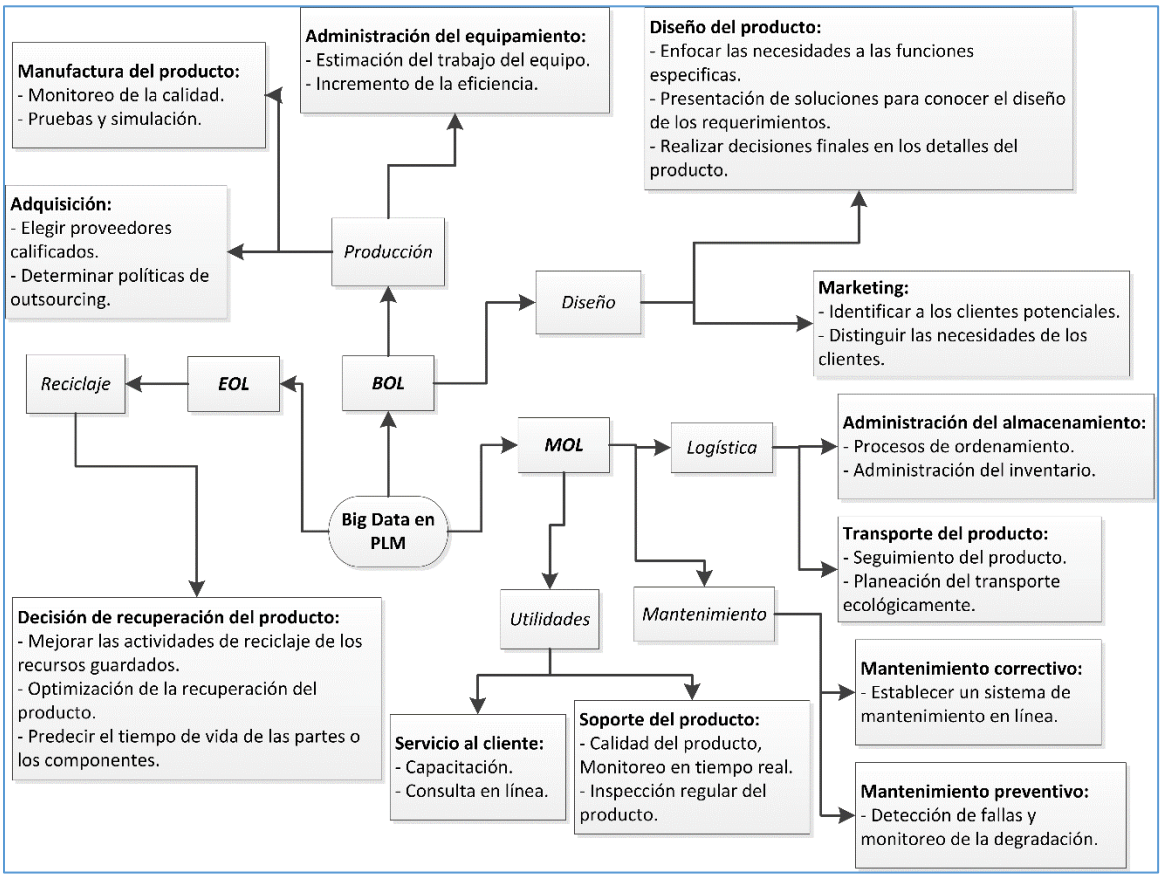


Ilustración 2. Framework de Big Data en PLM. (Li, Fei, Ying, & Liangjin, 2014)

1.5 Análisis de sentimiento

Martínez Cámara de la Red Temática en Tratamiento de la Información Multilingüe y Multimodal de España, mencionan que una de las múltiples tareas de las que se ocupa el Procesamiento del Lenguaje Natural (PLN) se encuentra la clasificación de textos, que consiste en la asignación de un conjunto de categorías a una colección de documentos, resolviéndose de esta forma la clasificación objetiva de documentos (Martínez Cámara, Martín Valdivia, & Ureña, 2011).

Existe una gran cantidad de textos en el que el contenido subjetivo es lo más relevante y cuyo procesamiento, no debería limitarse a aplicar únicamente las técnicas de la clasificación de documentos. Ante esta necesidad de clasificar la orientación o la opinión que se expresan en los documentos, surge el área análisis de sentimientos (AS) o en inglés *sentiment analysis*.

El análisis de sentimientos trata de clasificar los documentos en función de la polaridad de la opinión que expresa su autor. Esta nueva área que combina PLN y minería de textos, incluye una gran cantidad de tareas que han sido tratadas en mayor o menor medida. Existen principalmente dos formas distintas de enfrentarse a este problema: aplicando aprendizaje automático o aplicando un enfoque semántico. Dos son las aplicaciones más importantes: determinar la polaridad de las opiniones a nivel de documento, frase o característica y determinar si un documento contiene opiniones.

Existen muchos trabajos en el campo del análisis de sentimientos, habiéndose aplicado en multitud de dominios, pero la mayor parte de ellos han sido realizados sobre corpus, el cual es el conjunto finito de enunciados escritos o registrados, constituido para su análisis lingüístico, de documentos en inglés.

Richard Socher de la Universidad de Stanford, mencionan que los *espacios vectoriales semánticos* para las palabras han tenido uso ampliamente como herramienta, pero dado que ellos no pueden capturar el significado de frases largas apropiadamente, la composición en espacios vectoriales semánticos recientemente han tenido mucha atención. Sin embargo, el progreso ha sido frenado por la falta de fuentes compuestas calificadas y modelos, para capturar con precisión el fenómeno subyacente en tales datos. Para atender esta necesidad, introdujeron el *Stanford Sentiment Treebank and a powerful Recursive Neural Tensor Network*, que puede predecir asertivamente los efectos semánticos composicionales presentes en este nuevo corpus (Socher, y otros, 2013).

El *Stanford Sentiment Treebank* es el primer corpus completamente de árboles que permite un análisis completo de los efectos composicionales del sentimiento en el lenguaje. El cuerpo está basado en un conjunto de datos introducido por Pang y Lee en el 2005 en un artículo llamado *seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.,* y consiste de 11,855 oraciones extraídas de opiniones de películas, las cuales fueron procesadas por el analizador de Stanford desarrollado por Klein and Manning en el 2003, e incluyendo un total de 215,154 frases únicas de todos los árboles analizados, cada uno evaluado por tres jueces humanos. Este nuevo conjunto de datos permitió analizar las complejidades del sentimiento y capturar el fenómeno lingüístico complejo.

En la Ilustración 3. Ejemplo del Recursive Neural Tensor Network., podemos ver un ejemplo de la *Recursive Neural Tensor Network*, prediciendo cinco clases de sentimiento de manera precisa,

yendo desde muy negativo hasta muy positivo (--,-,0,+,++), en cada nodo del árbol de análisis, y capturando la negación y su alcance en la oración.

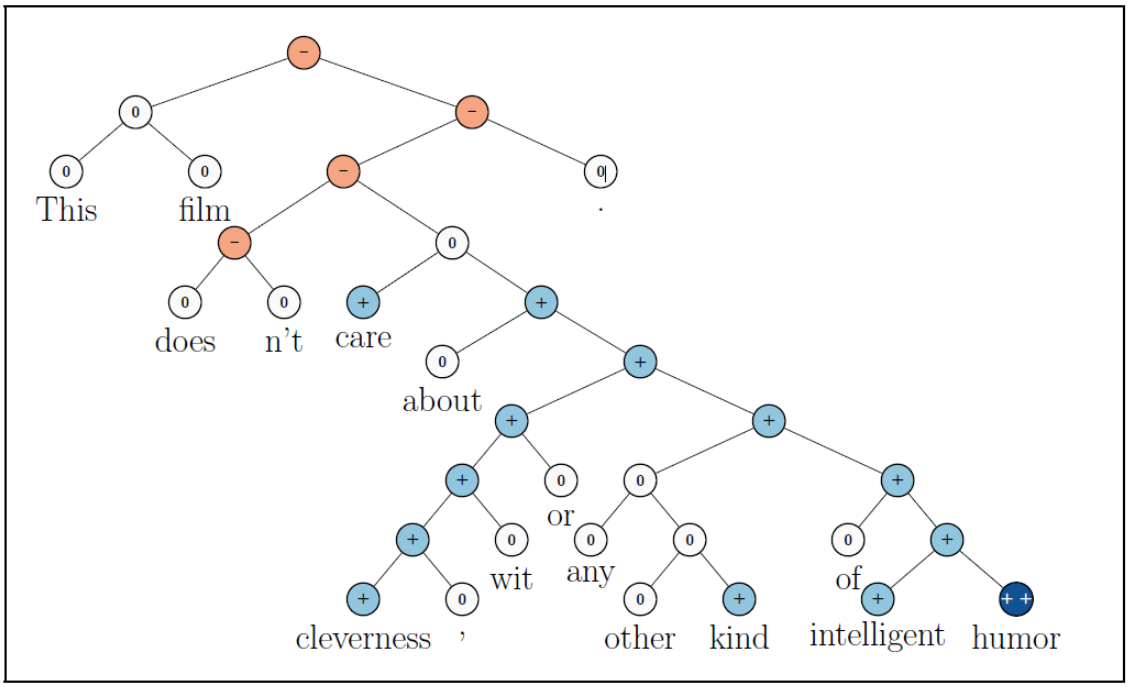


Ilustración 3. Ejemplo del Recursive Neural Tensor Network. (Socher, y otros, 2013)

1.6 *MapReduce*

MapReduce es un modelo de programación impulsado por Google que está asociado con implementaciones para el procesamiento y generación de grandes conjuntos de datos. El usuario especifica una función *map* que recibe parejas de tipo *key-value* para generar un conjunto intermedio de parejas del tipo *key-value,* para posteriormente utilizar la función *reduce,* que une todos los conjuntos intermedios con sus valores asociados con la misma llave del conjunto intermedio. La abstracción del modelo está inspirado en los primitivos *map* y *reduce* utilizados en los lenguajes funcionales como Lisp (Dean & Ghemawat, 2004).

1.6.1 ¿Por qué *MapReduce*?

Los sistemas empresariales tradicionales normalmente tienen servidores centralizados en donde almacenan y procesan sus datos. El modelo tradicional no está habilitado para procesar grandes volúmenes de datos y tampoco pueden distribuirla en servidores de bases de datos comunes. Además, los sistemas centralizados crean un cuello de botella por la operación de procesar múltiples archivos simultáneamente. Google resolvió el problema del cuello de botella utilizando el algoritmo de *MapReduce*, el cual divide un proceso en pequeños procesos y los distribuye en varias computadoras como se muestra en la Ilustración 4. Posteriormente, el resultado es recolectado en un solo lugar, e integrado para formar el resultado final del conjunto de datos.

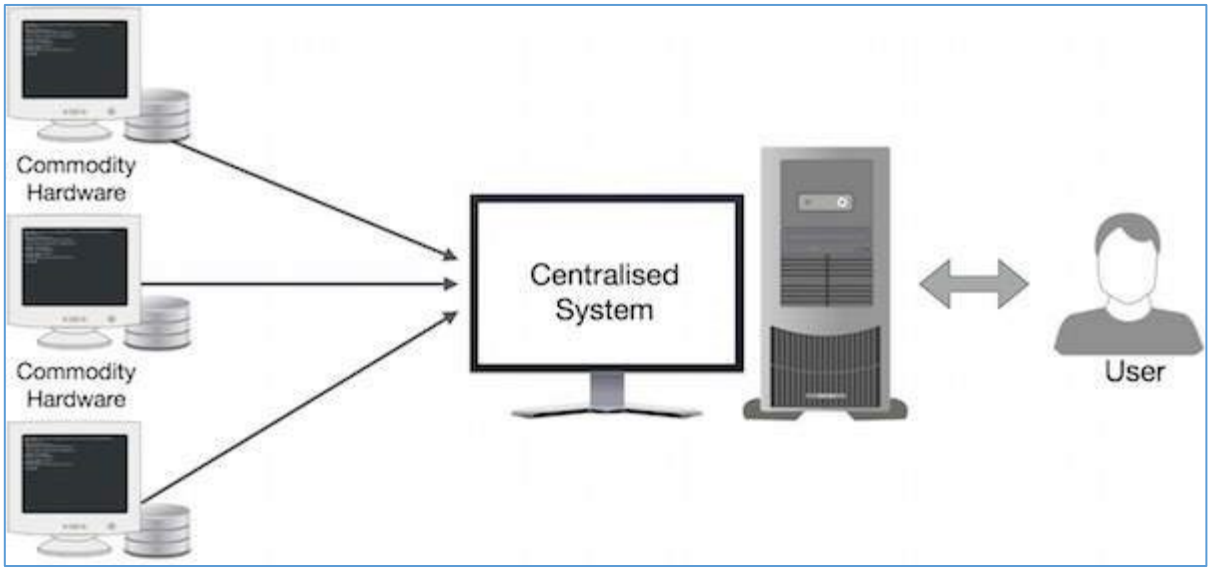


Ilustración 4. Distribución de MapReduce en el hardware. (Tutorialspoint, 2017)

1.6.2 ¿Cómo trabaja *MapReduce*?

El proceso de *map* toma un conjunto de datos y los convierte en otro, en donde los elementos son segmentados en tuplas del tipo *key*-value o llave-valor. El proceso de r*educe* toma la salida del proceso *map* como entrada y combina todas las tuplas de datos en conjuntos más pequeño.

Para ayudar a entender las dos tareas de *map* y *reduce*, se ejemplificará en la en donde vemos los datos de entrada en la fase de *Input*, se divide el conjunto de datos en la de fase *Split* para realizar el procesamiento de datos en paralelo, se mapean los datos en la *Map Phase*, se ordenan los resultados del mapeo en la fase *Shuffle and Sort* y por último se hace más pequeño el conjunto en la *Reduce Phase*.

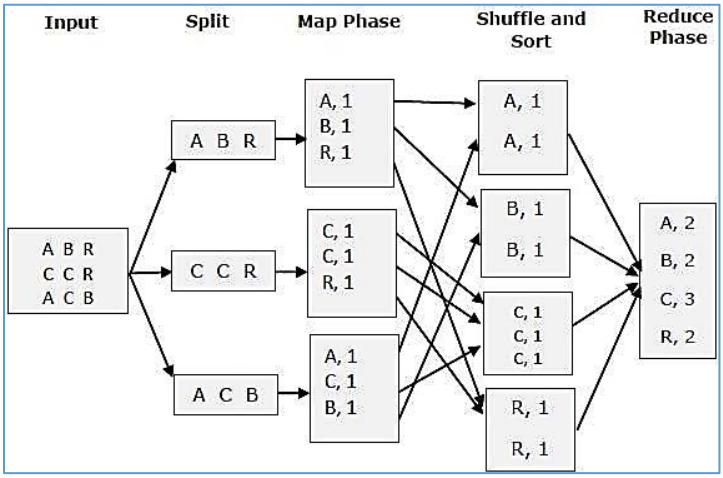


Ilustración 5. Funcionamiento del algoritmo MapReduce (Tutorialspoint, 2017).

Observando con mayor detenimiento cada fase, se podrá entender el significado de cada una de ellas que se muestran en la Ilustración 6.

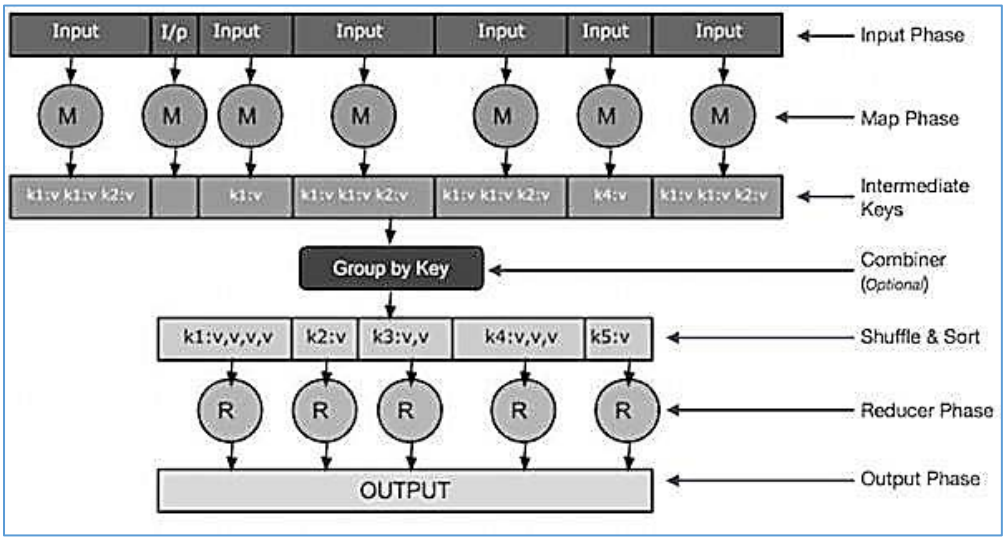


Ilustración 6. Fases del proceso MapReduce (Tutorialspoint, 2017).

Input: En esta fase, tendremos un lector de registros que transforma cada registro en un archivo de entrada y los datos transformados son enviados al *mapeador* en tuplas del tipo *key-value*.

Map: Es una función de usuario definida, la cual toma una serie de tuplas y las procesa cada una para generar cero o más tuplas.

Intermediate Keys: Las tuplas generadas por el mapeador ahora son conocidas como llaves intermedias.

Combiner: Un *combiner* es un tipo de reductor local el cual agrupa datos similares desde la fase de *map* en conjuntos identificables. Este toma las llaves intermedias del mapeador como entrada y aplica un código de usuario definido para agregar los valores en un mapeador más pequeño (esta parte no está dentro del algoritmo de *MapReduce*, es solamente opcional).

Shuffle and Sort: El proceso *reducer* empieza con el paso de *Shuffle & Sort*. Este descarga las tuplas agrupadas en una maquina local, en donde el *reducer* se ejecuta. Las tuplas individuales son ordenadas por la llave en una enorme lista de datos. La lista de datos agrupa las llaves equivalentes juntas, para que así sus valores puedan ser procesados fácilmente por el *reducer*.

Reducer: El *reducer* toma las tuplas agrupadas como entrada y ejecuta la función de *reducer* en cada una de ellas. En esta fase los datos pueden ser agregados, filtrados y combinados en un gran número de maneras y esta requiere un amplio rango de procesamiento. Cuando la ejecución termina, esta nos regresará cero o más tuplas en el paso final.

Output: En la fase de salida tenemos un formateador de salida que transforma las tuplas finales de la función *reducer* y las escribe en un archivo usando el grabador de registros.

1.7 *Hadoop* Distributed File System (HDFS)

El origen de *Hadoop* proviene del artículo sobre *Google File System*, de donde se desprendió la investigación de Google llamada *MapReduce: Simplified Data Proceesing on Large Cluster*, el cual establece un nuevo paradigma de programación orientado al procesamiento de datos masivos en un gran número de Clusters. El desarrollo inició con el *Apache Nutch Project* pero posteriormente fue movido al sub-proyecto de *Hadoop* en enero del 2006. En abril de 2006 fue lanzada la versión de *Hadoop* 0.1.0 y continúa su desarrollo a través de muchos contribuidores del *Apache Hadoop Project* (Hortonworks, 2016).

En el 2011, Rob Bearden se asoció con Yahoo! para crear Hortonworks de la mano de 24 ingenieros del proyecto original de Hadoop, incluyendo a los fundadores del *Apache Hadoop Project* Alan Gates, Arun Murthy, Devaraj Das, Mahadev Konar, Owen O'Malley, Sanjay Radia, y Suresh Srinivas (Hortonworks, 2016).

1.7.1 Beneficios

Algunas de las razones por las cuales las organizaciones usan *Hadoop* es la capacidad de almacenar, administrar y analizar enormes cantidades de datos estructurados y no-estructurados de manera rápida, confiable, flexible y a un bajo costo.

1. Escalabilidad y performance: El procesamiento distribuido en cada nodo de manera local con *Hadoop* para el almacenamiento, procesamiento y análisis de los datos en escalas de peta bytes.
2. Confiabilidad: El cómputo en grandes cantidades de nodos es propenso a fallas individuales de los mismos. *Hadoop* es fundamentalmente resistente, cuando un nodo llega a fallar este es re-direccionado a otro nodo en el clúster y los datos son automáticamente replicados para la reparación de fallas futuras en los nodos.
3. Flexibilidad: A diferencia de los tradicionales manejadores de bases de datos, no se requiere crear esquemas estructurados antes de almacenar los datos. Se pueden almacenar datos en cualquier formato incluyendo semi-estructurado y no estructurado para después ser leídos, estos son procesados para aplicarles un esquema a los datos.
4. Bajo costo: A diferencia del software propietario, *Hadoop* es software libre y funciona con hardware básico de bajo costo (Hortonworks, 2016).

1.7.2 Arquitectura

Se ha desarrollado utilizando el diseño de *Sistema de Archivos Distribuidos de Hadoop* (HDFS), componiéndose de un *NameNode* que contiene la descripción y los atributos de los datos, el *DataNode* que son los contenedores de datos y un cliente HDFS que realiza las peticiones de consulta. Se ejecuta en productos de hardware básicos. A diferencia de otros sistemas distribuidos, el HDFS es muy tolerante y diseñado para utilizar hardware de bajo costo.

Los HDFS tienen una gran cantidad de datos y proporciona un acceso más eficiente. Para almacenar estos datos de gran tamaño, los archivos HDFS se almacenan en varias máquinas. Estos archivos se almacenan en forma redundante para recuperar el sistema de posibles pérdidas de datos en caso de fallo. HDFS también permite aplicaciones de procesamiento en paralelo (Konstantin, Hairong, Sanjay, & Robert, 2010).

A continuación se mencionan las características más importantes de los HDFS:

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

1.7.2.1 NameNode

El espacio de trabajo HDFS es una jerarquía de archivos y directorios. Los archivos y directorios son representados en el *NameNode* por *inodes*, los cuales guardan atributos como permisos, tiempo de modificación y acceso, espacios de trabajo y los límites de almacenamiento en los discos. El contenido de un archivo es dividido en bloques grandes (comúnmente de 128 megabytes), y cada bloque del archivo es replicado independientemente en múltiples *DataNodes* (comúnmente en tres). El *NameNode* contiene el árbol de los espacios de trabajo y el mapeo de la ubicación física de los bloques de los archivos contenidos en los *DataNodes*.

Un cliente que quiera leer un archivo, primero contacta al *NameNode* para localizar los bloques de los datos para ser leídos por el *DataNode* más cercano al cliente. Cuando un cliente quiere escribir datos, el cliente solicita al *NameNode* que seleccione tres *DataNodes* para almacenar las réplicas de los datos. El cliente posteriormente escribe los datos en los *DataNodes* en una secuencia de tipo *pipeline*. Este diseño de la arquitectura tiene un *NameNode* por cada clúster. Un clúster puede tener cientos de *DataNodes* y cientos de miles de clientes HDFS. Cada *DataNode* puede ejecutar tareas de aplicaciones de manera concurrente (Konstantin, Hairong, Sanjay, & Robert, 2010).

1.7.2.2 DataNode

Cada réplica de un bloque de datos en el *DataNode* es representado por dos archivos dentro del sistema de archivos nativo del sistema. El primer archivo contiene los datos mismos y el segundo archivo contiene los metadatos del bloque de datos. El tamaño del archivo de datos es igual a la contenida en el bloque y no requiere espacio extra aunque el bloque esté a la mitad, en el sistema de archivos nativo únicamente ocupará lo contenido dentro del bloque. Esto quiere decir que si un bloque es ocupado a la mitad, aunque este sea de 128 megabytes, únicamente ocupará 64 megabytes en el sistema de archivos nativo del sistema operativo (Konstantin, Hairong, Sanjay, & Robert, 2010).

1.7.2.3 Cliente HDFS

Las aplicaciones de los usuarios ingresan al sistema de archivos usando el cliente HDFS, el cual es una biblioteca que funge como interface del HDFS.

Similar a los sistemas de archivos convencionales, el HDFS soporta operaciones de lectura, escritura y borrado de archivos, y operaciones de creación y borrado de directorios. El usuario referencia archivos y directorios desde rutas en los espacios de trabajo. Es muy importante saber que las aplicaciones del usuario, generalmente no necesitan saber que los metadatos del sistema de archivos y el almacenamiento, están almacenados en diferentes servidores o que los bloques tienen múltiples réplicas.

Cuando una aplicación lee un archivo, el cliente HDFS primero pregunta al *NameNode* por la lista de *DataNodes* que contienen las réplicas de los bloques de los archivos. Posteriormente el *NameNode*

contacta al *DataNode* directamente y solicita la transferencia del bloque deseado. Cuando un cliente escribe, primero pide al *NameNode* que escoja *DataNodes* para almacenar las réplicas del primer bloque del archivo. El cliente organiza un *pipeline* de nodo a nodo y envía los datos. Cuando el primer bloque es enviado completamente, el cliente solicita nuevos *DataNodes* a ser elegidos para almacenar el siguiente bloque de la réplica. Un nuevo pipeline es organizado y el cliente envía los datos de los archivos. Cada selección de *DataNodes* es diferente. La Ilustración 7 muestra la interacción del *NameNode*, el *DataNode* y el cliente (Konstantin, Hairong, Sanjay, & Robert, 2010).

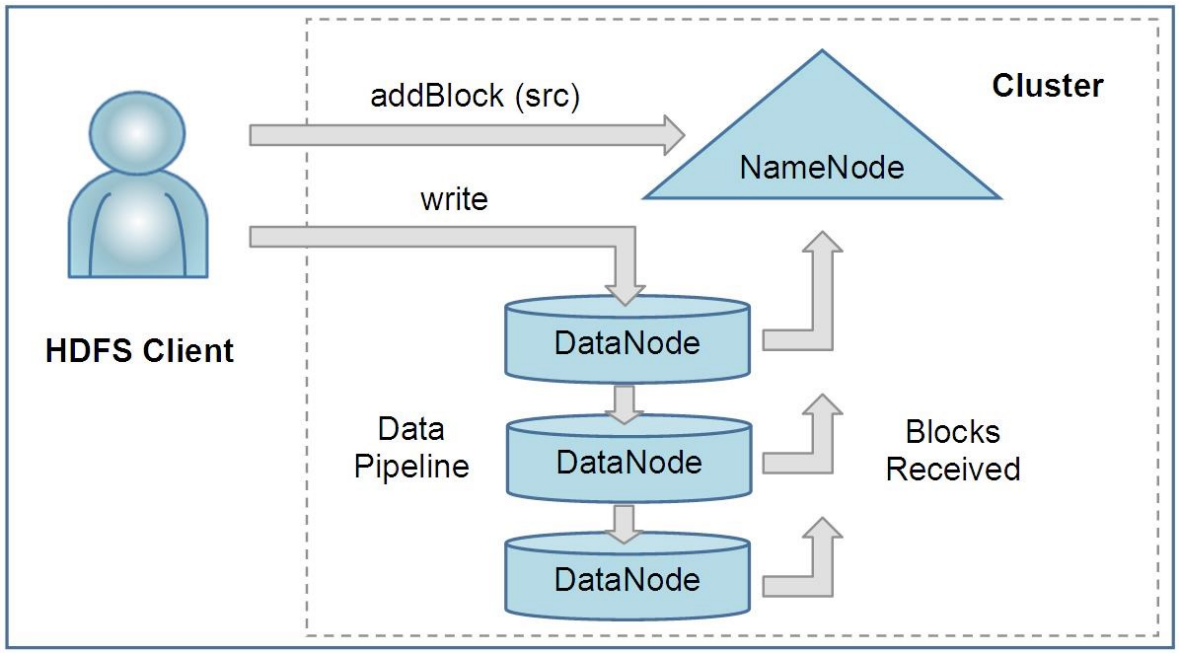


Ilustración 7. Interacción entre el NameNode, el DataNode y el cliente. (Konstantin, Hairong, Sanjay, & Robert, 2010)

1.8 *Hive*

Hadoop no es fácil para el usuario final, especialmente para aquellos que no están familiarizados con el paradigma de *MapReduce*. Anteriormente los usuarios tenían que escribir programas basados en *MapReduce* para tareas simples como contar los resultados obtenidos u obtener promedios. *Hadoop* carece de la expresividad como la tiene *SQL*, por lo que el usuario final pasaba de horas a días escribiendo programas para obtener análisis simples. En enero de 2007 el equipo de infraestructura de datos de Facebook tomó la iniciativa de crear *Hive*, con su visión de tomar conceptos familiares como las tablas, columnas, particiones y subconjuntos de *SQL* para el ambiente sin estructura de Hadoop, manteniendo todas las ventajas que este tiene. *Hive* se convirtió en código abierto en agosto de 2008.

La estructura de datos que maneja *Hive* se basa en los conceptos de una base de datos, como lo son las tablas, columnas, filas y particiones. Soporta todos los tipos de datos primitivos como son los enteros, flotantes, doubles y strings, como también tipos complejos como mapas, listas y estructuras de datos. Adicionalmente *Hive* permite al usuario extender el sistema con sus propios tipos de datos y funciones. El lenguaje *HiveQL* o *HQL* de consulta es muy similar a *SQL* por lo que puede ser fácilmente entendido por alguien familiarizado con él (Thusoo, y otros, 2010).

1.9 *Pig Latin*

Pig Latin es un lenguaje que combina el alto nivel de las declaraciones de consulta en la esencia de *SQL* y el bajo nivel de la programación procedural del estilo de *MapReduce*. Para entender mejor el concepto, tomaremos como ejemplo que tenemos una tabla que contiene direcciones web: (*url, category, pagerank*). La siguiente declaración en *SQL* nos devuelve por cada categoría suficientemente grande, el promedio del ranking de cada una.

```
SELECT category, AVG(pagerank)
FROM urls WHERE pagerank > 0.2
GROUP BY category HAVING COUNT(*) > 10^6
```

Su equivalente en *Pig Latin* es el siguiente:

```
good_urls = FILTER urls BY pagerank > 0.2;
groups = GROUP good_urls BY category;
big_groups = FILTER groups BY COUNT(good_urls)>10^6;
output = FOREACH big_groups GENERATE
        category, AVG(good_urls.pagerank);
```

Se puede notar que el programa en *Pig Latin* es una secuencia de pasos, muy parecido a un lenguaje de programación y por cada paso se obtiene una simple transformación de los datos. Al mismo tiempo, la transformación llevada a cabo en cada paso es de alto nivel, por ejemplo los filtros, las agrupaciones y las funciones de agregación son muy similares en *SQL*. Se detallarán el uso de las sentencias en el capítulo 4 en la sección 4.

En efecto, escribir un programa en *Pig Latin* es similar a especificar un plan de ejecución de consultas, de esta manera a los programadores se les facilita entender y controlar la ejecución de las tareas para el procesamiento de los datos (Olston, Reed, Srivastava, Kumar, & Tomkins, 2008).

1.10 *Storm*

La década pasada el procesamiento de datos tuvo una revolución. *MapReduce*, *Hadoop* y tecnologías relacionadas, hicieron posible almacenar y procesar datos a escalas antes inimaginables. Desafortunadamente, esas tecnologías de procesamientos de datos no era sistemas en tiempo real y no había manera de hacer con *Hadoop* un sistema en tiempo real. Un sistema de procesamiento de datos en tiempo real contiene un conjunto de diferencias fundamentales de requerimientos que uno de procesamiento por lotes. A pesar de todo, el procesamiento de datos en tiempo real en escala masiva se convirtió en un requerimiento constante para los negocios. La falta de un "*Hadoop* en tiempo real" se convirtió en una gran brecha en el ecosistema de procesamiento de datos, por lo que *Storm* llenó esa brecha (Apache Software Foundation Storm, 2016).

Apache Storm es un sistema de cómputo en tiempo real distribuido, libre y de código abierto. *Storm* hace fácil y fiable el procesamiento de flujos de datos continuos, haciendo en tiempo real el procesamiento que hace *Hadoop* por lotes. *Storm* es simple y puede ser usado con cualquier lenguaje de programación.

Storm tiene múltiples usos como por ejemplo: análisis en tiempo real, aprendizaje automatizado en línea, cómputo continuo, RCP distribuido, *ETL*, y más. *Storm* es rápido, se le cronometró más de un millón de tuplas procesadas por segundo por nodo. Es escalable, tolerante a fallas, garantiza que los datos serán procesados y fácil de configurar y operar.

1.11 *NiFi*

Apache NiFi fue desarrollado para automatizar y administrar el flujo de información entre sistemas. Este problema se presenta al momento que las empresas empiezan a tener más de un sistema, donde uno de los sistemas crea los datos y otro los consume (Apache Software Foundation NiFi, 2016).

Algunos de los principales retos para administrar los flujos de datos son por ejemplo:

- Fallas en la red.
- Fallas en los discos.
- Caídas del software.
- Errores humanos.
- El acceso a los datos excede la capacidad de consumo.
- Reorganizar y priorizar rápidamente para activar nuevos flujos de datos.
- Cambiar a esquemas más rápidos.
- Adaptarse al cambio de infraestructura.
- Seguridad.

Al paso de los años, el flujo de datos ha sido uno de los males necesarios en las arquitecturas. Ahora en día hay una gran variedad de opciones para manejar flujos de datos efectivamente. Se incluyen opciones como arquitecturas orientadas a servicios (SOA), las implementaciones tipo *API*, internet de las cosas (IoT, por sus siglas en inglés), y el *Big Data*. La principal diferencia entre ellas son los objetivos y la complejidad de la tarea, el grado de los retos necesarios para adaptarse, etc. *NiFi* está desarrollado para ayudar a atacar los retos modernos sobre los flujos de datos.

1.11.1 Arquitectura *NiFi*

En la Ilustración 8 se muestra la arquitectura de *Apache NiFi*.

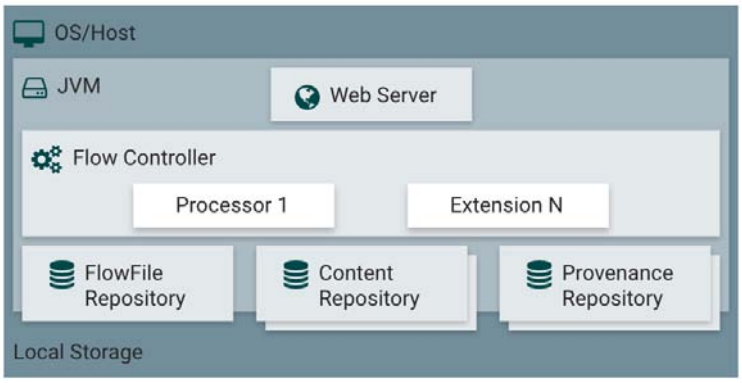


Ilustración 8. Arquitectura NiFi (Apache Software Foundation NiFi, 2016).

NiFi se ejecuta sobre la máquina virtual de java contenida en un sistema operativo. Los componentes principales son los siguientes:

- Servidor web: El propósito del servidor web es contener el sitio basado en http de *NiFi* para controlar la *API*.
- Controlador de flujo: Es el cerebro de la operación. Provee hilos de ejecución y administra la programación de las tareas y las solicitudes de ejecución solicitadas por las extensiones.
- Repositorio del flujo de Archivos: El objetivo principal es tener un área persistente para la escritura de un log en una partición específica del disco.
- Repositorio de contenido: Es donde el contenido de bytes dados por el repositorio del flujo de archivos opera, almacenando bloques de datos en el sistema de archivos.
- Repositorio de procedencias: Es donde se guardan todas las procedencias de los datos almacenados.

Con la versión *NiFi* 1.0, se implementó el paradigma Zero-Master Clustering que se muestra en la Ilustración 9. Cada nodo en el cluster *NiFi* realiza las mismas tareas sobre los datos, pero cada uno opera sobre diferentes grupos de datos. *Apache* ZooKeeper es un administrador de clusters quien selecciona un nodo del cluster como coordinador y el Nodo Primario (Apache Software Foundation NiFi, 2016).

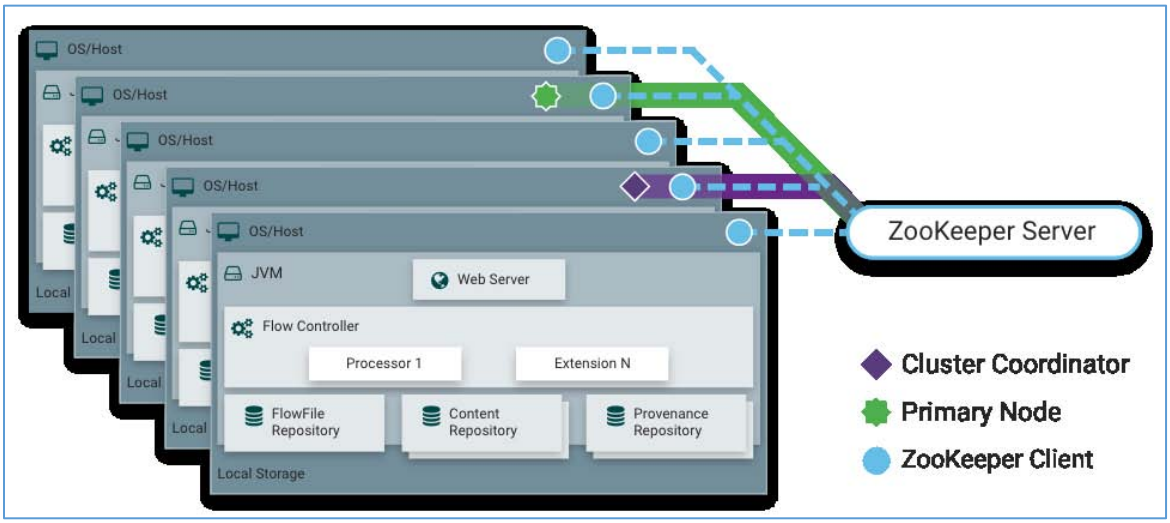


Ilustración 9. Paradigma. Zero-Master Clustering.

1.12 Kafka

Apache Kafka se define como una plataforma de flujo distribuida. Lo que significa que tiene tres ventajas:

1. Permite publicar o suscribirse a flujos de datos. Lo cual es similar a un proceso de mensajes o a un sistema de mensajes empresarial.
2. Permite almacenar flujos de datos de una manera tolerante a fallos.
3. Permite procesar flujos de datos en el momento que estos ocurran.

Entonces Kafka es bueno para dos clases de aplicaciones:

1. Construir aplicaciones de flujo de datos de tipo pipeline en tiempo real, que obtenga de manera confiable los datos entre el sistema o las aplicaciones.
2. Construir aplicaciones que flujos de datos en tiempo real, que transformen o reaccionen a los datos del flujo.

Como podemos ver su arquitectura en la Ilustración 10, es una plataforma de flujo distribuido. Funciona como cluster en uno o más servidores. El cluster de Kafka almacena flujos de registros en categorías llamadas *topics* o temas. Cada tema consiste en una llave, un valor y su *timestamp* (Apache Software Foundation KafKa, 2016).

Kafka tiene cuatro *API's* principales.

1. La *API* productora: Permite a una aplicación publicar un *stream* de registros a uno o más *topics*.
2. La *API* consumidora: Permite a una aplicación subscribirse a uno o más topics y procesar el stream de registros producidos en ellos.
3. La *API* de *streams*: Permite a una aplicación actuar como un procesador de *stream*, consumiendo la entrada de un stream desde uno o más topics y produciendo un *stream* de salida a uno o más topics de salida, transformando efectivamente la entrada del *stream* a la salida del stream.
4. La *API* conectora: Permite construir y correr productores o consumidores que conecten topics de Kafka a aplicaciones existentes o sistemas de datos. Por ejemplo: un conector a una base de datos relacional puede capturar cada cambio en ella en una tabla.

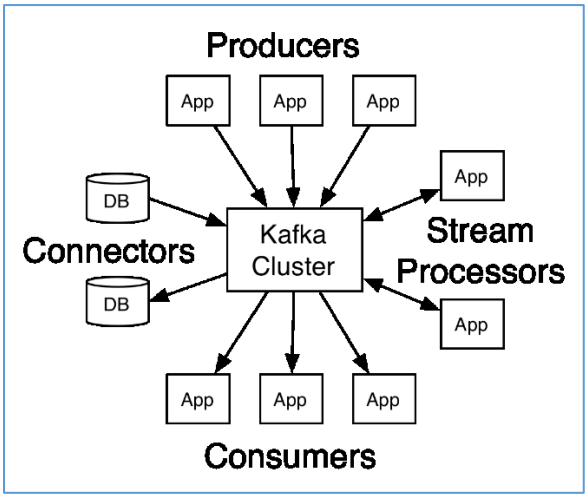


Ilustración 10. Arquitectura de Kafka. (Apache Software Foundation KafKa, 2016)

Kafka provee un cliente en Java y en varios lenguajes como C/C++, Python, Go (AKA golang), Erlang, .NET, Clojure, Ruby, Node.js, Proxy (HTTP REST, etc), Perl, stdin/stdout, PHP, etc.

1.13 Hortonworks Data Platform (HDP)

La arquitectura HDP es la única distribución de *Apache Hadoop* de código abierto y lista para las empresas basada en una arquitectura centralizada (YARN). HDP responde a las necesidades completas de los datos en espera, concentrando un gran número de herramientas de *Apache* para el procesamiento de datos, impulsa las aplicaciones en tiempo real de los clientes y ofrece un análisis robusto que acelera la toma de decisiones y la innovación (Hortonworks, 2016).

Hortonworks se comprende de los módulos que se muestra en la Ilustración 11, que se describen a continuación:

- Gobernanza de la integración *(Governance Integration)*: Engloba las herramientas necesarias para conectarse a fuentes de datos externas para integrar los datos al marco de trabajo.
- Herramientas: Contiene las interfaces de usuario necesarias para administrar y visualizar la configuración del marco de trabajo y los datos resultantes del procesamiento de los datos.
- Acceso a los datos *(Data Access)*: Contiene las aplicaciones necesarias para realizar consultas a los datos almacenados, procesarlos, transformarlos, analizarlos y operarlos, según sea el caso y la necesidad del proyecto.
- Administración de los datos *(Data Management)*: Contiene la base de datos almacenada en archivos HDFS.
- Seguridad *(Security)*: Contiene los programas necesarios para administrar el acceso de los usuarios a los contenedores de datos.
- Operadores *(Operations)*: Contiene las herramientas necesarias e interfaces para administrar y monitorear el servidor que contiene el marco de trabajo.

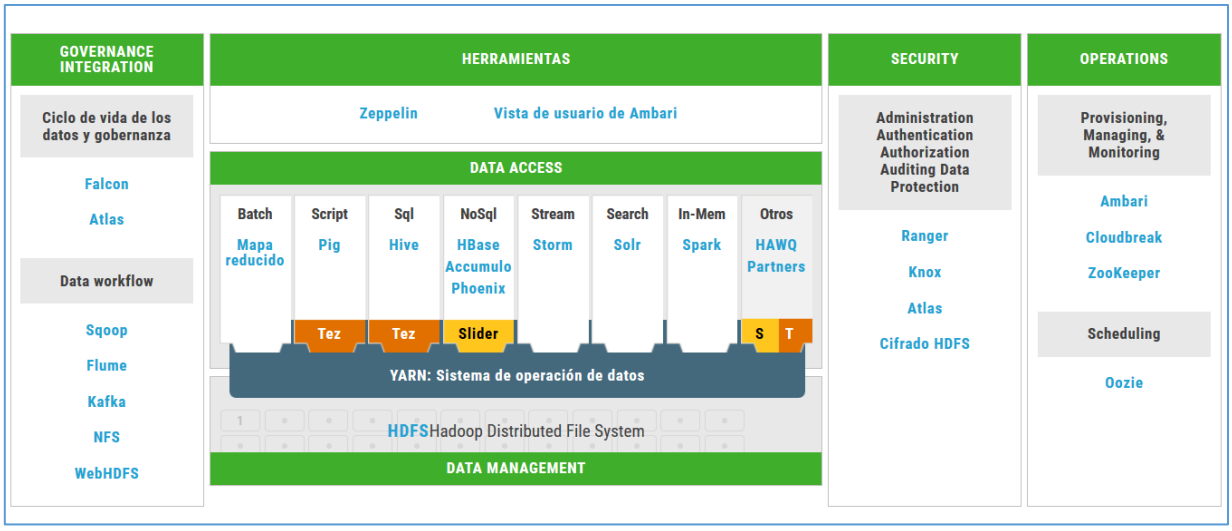


Ilustración 11. Arquitectura HORTONWORKS DATA PLATFORM (HDP) (Hortonworks, 2016)

Capítulo 2. Antecedentes

El presente capítulo tiene como objetivo, el justificar cómo mediante datos provenientes de las redes sociales como fuente de datos masivos, se realizan investigaciones desde el sector salud, hasta las marcas comerciales. Se abordarán los objetivos y qué métodos se implementaron, como el análisis de sentimiento y el desarrollo de proyectos de *Big Data*.

Las redes sociales, también conocidas como *social media* en el área comercial, han incrementado su uso significativamente en los años recientes, lo cual ha levantado el interés de investigadores para realizar estudios sobre el comportamiento de los usuarios, detectando los llamados *trending topics*, que básicamente son las apariciones de los *hashtags* que crean los usuarios para referirse a un tema de su interés.

El *Pew Research Center* publicó un artículo sobre el uso de las redes sociales en Estados Unidos de Norteamérica, en donde afirma que, en la población se está utilizando una amplia variedad de fuentes de información y maneras de interactuar con otras personas. Por ejemplo, la mayoría de las personas recurrieron a las redes sociales para obtener noticias, y la mitad del público utilizó estos medios para informarse sobre las elecciones presidenciales del 2016. Por otra parte, la población está utilizando las redes sociales en el ambiente laboral, por poner un ejemplo, para distracción y relajación en horas laborales, o bien, para buscar un nuevo empleo. En una encuesta nacional donde entrevistaron a 1520 adultos, entre el 7 de marzo y 4 de abril de 2016, se encontró que Facebook continúa siendo la red social más popular en un margen bastante amplio, quedando la cifra que un 79% de adultos son usuarios de Facebook, 32% de Instagram, 31% de Pinterest, 29% de LinkedIn y 24% de Twitter. (Greenwood, Perrin, & Duggan, 2016)

En el artículo de Chandler McClellan, estima que el 18% de los adultos de E.U.A. tienen una cuenta de twitter. El servicio de twitter permite a los usuarios publicar mensajes breves a no más de 140 caracteres de longitud. La naturaleza pública de esta red social, comparándolas con otras en la que las personas mantienen abiertos sus perfiles para ser consultados por todos los usuarios, la convierte en una fuente importante de información, ya que tanto personas como organizaciones hacen públicas sus opiniones en una gran variedad de temas (McClellan, Ali, Mutter, Kroutil, & Landwehr, 2016).

2.1 Estudios sobre el análisis en redes sociales

El uso de las redes sociales por un gran número de usuarios, incrementa el interés de los investigadores para realizar estudios de datos masivos. Para dichas investigaciones, se requiere aplicar técnicas, métodos y herramientas, para la extracción y análisis de los datos en las redes sociales como se verá a continuación.

2.1.1 Estudio sobre la variabilidad temporal del problema del alcoholismo en twitter

En el 2012 la *Open Journal of Preventive Medicine*, publicó un artículo referente a detectar las menciones en el tiempo sobre el consumo de bebidas alcohólicas, para así saber en qué momento del día y de la semana se refleja su consumo en redes sociales. (West, y otros, 2012)

En este estudio, se planteó como objetivo el medir el índice de aparición de menciones, referentes al consumo de bebidas alcohólicas en ciertos periodos de tiempo, proponiendo la Teoría del Comportamiento Planificado. En el estudio se propone un marco, en donde se relacionan los contenidos de twitter, con los comportamientos de sus usuarios, para identificar el problema del alcoholismo.

La Teoría del Comportamiento Planificado, estudia cómo el comportamiento de unas personas puede influir con otras, ya que están expuestas a recibir mensajes u opiniones, los cuales pueden ser interpretados como comportamientos aceptados, como lo es el consumo de alcohol, sobre todo, si el mensaje proviene de una persona considerada importante.

Hoy en día, a las personas que promueven marcas, como productos, servicios, etc., en sus redes sociales, se les conoce como *Influencers*.

En dicho artículo se explica cómo se implementó un método para el objetivo general, el cual consistió de los siguientes pasos:

1. Se utilizó la Twitter Application Programming Interface (*API*) con la cual obtuvo 5,697,008 de tuits generados de nueve estados de la unión americana seleccionados con base en las diez ciudades más pobladas de la Unión Americana.
2. Utilizando la opción de búsqueda de la *API*, consultando los tuits más recientes generados por cada estado en un intervalo de 2 minutos, en un periodo de tiempo de 31 días que comprendieron de 5 de octubre de 2010 al 3 de noviembre de 2010, y otro periodo de 5 días, a partir del 30 de diciembre de 2010 al 3 de enero de 2011, esto con el objetivo de medir las menciones en una fecha especial como lo es la celebración del año nuevo.
3. Todos los tuits fueron importados a una base de datos para ser analizados utilizando el software de SAS, el cual para obtener los resultados estadísticos, se basó en la frecuencia de aparición de las palabras definidas.
4. Todas las palabras que no estuvieran en el idioma inglés fueron excluidas.
5. Se buscaron palabras relacionadas con el problema del alcohol como *drunk* y menciones sobre bebidas como *beer*.
6. Se utilizó una compilación de palabras comúnmente utilizadas en el tema del alcoholismo y sinónimos, como por ejemplo, sinónimos de la palabra *drunk: wasted, tipsy, intoxicated, hammered, sauced, buzzed, trashed, etc.*
7. Se utilizó la capacidad de geolocalización de la *API*, para obtener los tuits de la región definida sobre las ciudades más pobladas, de acuerdo al censo de población.

Es bien sabido que el consumo de bebidas alcohólicas se da principalmente los fines de semana y en fechas especiales, por lo que este estudio se enfocó en la detección de este patrón de comportamiento en twitter. (West, y otros, 2012)

La conclusión que resultó fue que las menciones aparecían en los fines de semana y en fechas especiales como se aprecia en la Ilustración 12.

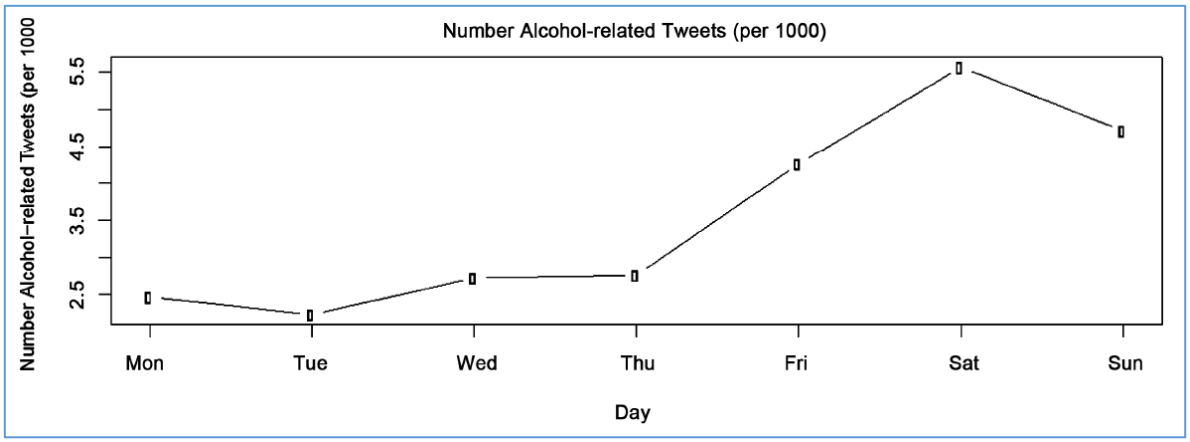


Ilustración 12. Número menciones relacionadas con el consumo de alcohol en los días de la semana. (West, y otros, 2012)

2.1.2 Explorando twitter sobre el consumo de drogas psicoestimulantes entre los estudiantes

A continuación se muestra el uso de redes sociales para la detección de patrones de consumo de una droga llamada Adderall entre los estudiantes de diversas universidad de E.U.A., tomando como fuente de datos twitter. (Hanson, y otros, 2013)

En un artículo publicado por la *Journal of Medical Internet Research*, se aplicó un estudio sobre el consumo de Adderall, que es comúnmente prescrita para el tratamiento sobre el déficit de atención y desorden hiperactivo. Esta droga es muy popular en su consumo por los estudiantes de acuerdo a la encuesta nacional sobre el consumo de drogas y salud, 6.4% de los estudiantes entre 18 y 22 años han consumido Adderall en el año 2012. Esta droga se popularizó principalmente por que ayudaba a los estudiantes en momentos de estrés, a poder tener una mayor concentración e incrementar su lucidez mental.

Carl L Hanson, menciona que el uso de las redes sociales provee una nueva y poco explorada fuente, para el monitoreo y entendimiento de los problemas de salud pública. De igual manera, afirma que la capacidad de obtener en tiempo real datos de las redes sociales, hace mucho más rápido un análisis comparado con las herramientas tradicionales, como lo son las encuestas y los cuestionarios. También menciona que los estudios han demostrado la utilidad de la información en línea, para entender los problemas de salud pública y sus causantes. Como por ejemplo, datos obtenidos por las tendencias en las búsquedas en internet, predijeron brotes de influenza, listeriosis de comida contaminada, gastroenteritis y varicela.

Se planteó como objetivo:

1. Detectar menciones relacionadas con Adderall en twitter.
2. Identificar la variación en la cantidad de menciones en periodos de exámenes.
3. Detectar las diferencias en el número de menciones entre colegios y universidades.
4. Detectar los efectos comunes de la droga
5. Y con que otras substancias se consume.

En el estudio, se realizaron las siguientes actividades:

1. Se tomaron los colegios y universidades con más de 10,000 estudiantes.
2. Se utilizó el *API* de twitter para obtener los datos aprovechando la búsqueda por geolocalización.
3. Se aplicó el Agrupamiento Aglomerado Jerárquico.
4. Se creó un catálogo de palabras relacionadas y sinónimos de las mismas para realizar la búsqueda.
5. Los datos fueron exportados a una hoja de cálculo de Microsoft Excel para posteriormente ser importada al software de SPSS versión 20 para el análisis.
6. Frecuencias, porcentajes, media, mediana y desviación estándar fue utilizada para describir el abuso de Adderall.
7. ArcGIS 10 fue usado para crear mapas con porcentajes sobre ubicaciones específicas dadas por el GPS.

Se obtuvieron 14,282 tuits de personas que contenían la palabra Adderall o pharm en sus nombres de usuario, los cuales fueron removidos ya que no se consideran usuarios típicos, más bien, se aceptaban todos aquellos quienes en sus publicaciones mencionaban la palabra Adderall. La

muestra considerada fue de 213,633 tuits que mencionaban Adderall, de 132,099 cuentas de usuarios únicos.

La vasta mayoría de los tuits que contenían la palabra relacionada con Adderall, eran bromas o sarcasmo, lo cual dificultaba obtener un análisis preciso. Sin embargo, continuando con el estudio, se observó que las palabras relacionadas con la droga, tenía mayor afluencia entre semana, teniendo un pico en miércoles como lo muestra la Ilustración 13.

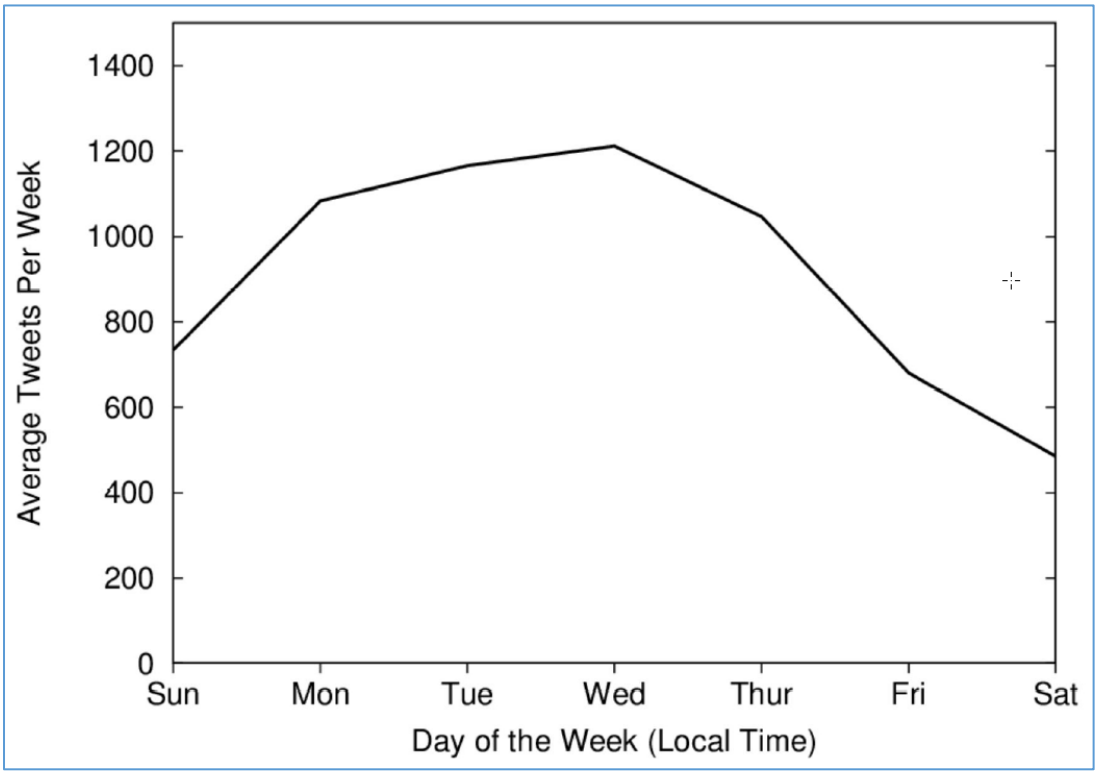


Ilustración 13. Tuits relacionados con adderall por día de la semana. (Hanson, y otros, 2013)

Por otra parte, podemos observar en la Ilustración 14, el número de tuits por día variaba significativamente a lo largo del año, siendo constante una mayor presencia en los días entre semana que en los fines de semana. También se observaron picos en los meses de diciembre y mayo, que son los meses tradicionalmente de exámenes.

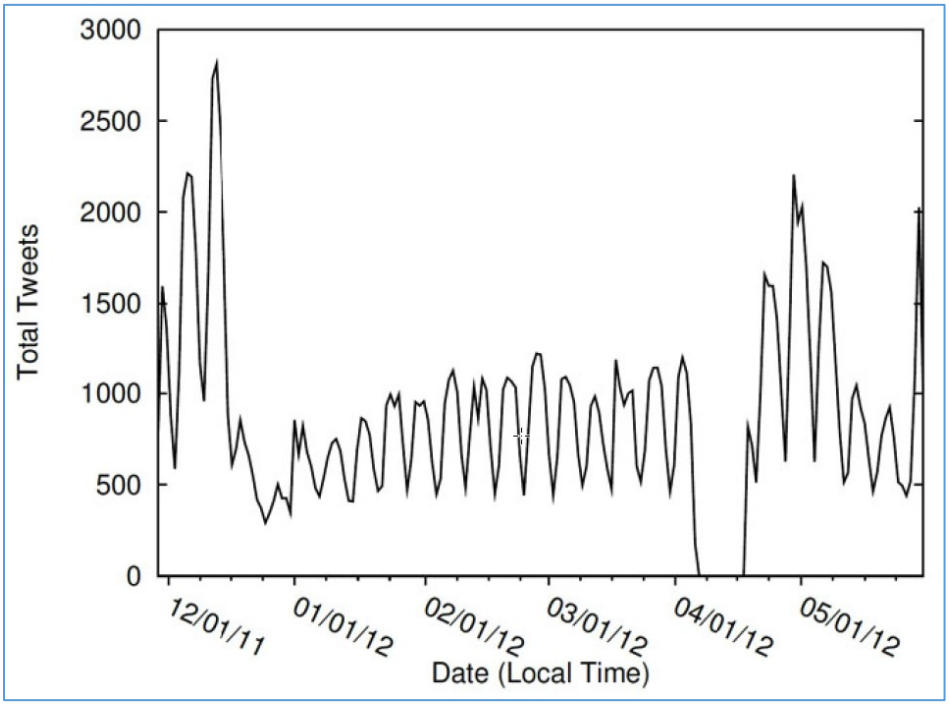


Ilustración 14. Distribución de Adderall en seis meses. (Hanson, y otros, 2013)

Los resultados en cuanto a la cantidad de menciones, se observó que se tiene mayor presencia en el noroeste y sur entre los colegios y universidades como se muestra en la Ilustración 15. La principal sustancia relacionada con las menciones de Adderall fue el alcohol seguido de otros estimulantes, y el efecto más mencionado fue la inhibición del sueño, seguido de la pérdida del apetito.

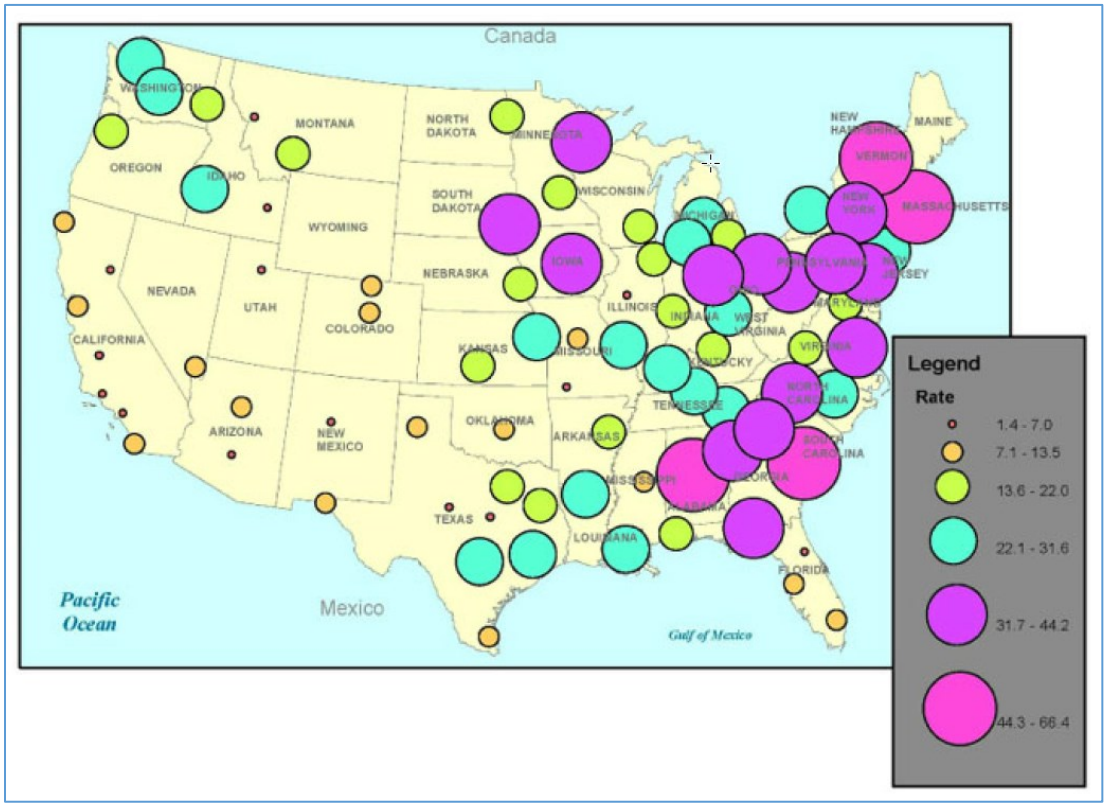


Ilustración 15. Presencia de adderall en un radio de 150 millas alrededor del colegio o universidad. (Hanson, y otros, 2013)

2.1.3 Utilizando las redes sociales para monitorear las discusiones sobre salud mental

En el año 2016, la *Journal of the American Medical Informatics Association*, realizó un análisis sobre las menciones referentes a la depresión y suicidio entre los usuarios de twitter, en el cual publicó un artículo sobre el uso de las redes sociales para monitorear la salud mental. (McClellan, Ali, Mutter, Kroutil, & Landwehr, 2016)

Su objetivo era poder identificar periodos de intensificación en el interés sobre los temas de salud mental en twitter, enfocándose en el suicidio y la depresión.

Para el estudio sobre el monitoreo de la salud mental, se realizaron las siguientes actividades:

1. Se utilizó el Crimson Hexagon's ForSight Software, el cual provee acceso a las redes sociales públicas como twitter, del cual se obtuvieron 176 millones de tuits del 2011 al 2014 para términos relacionados con el suicidio y la depresión.
2. Del sitio http://hashtagify.me/ se obtuvieron los *hashtags* relacionados con el tema de la depresión y el suicidio.
3. En la obtención de los tuits relacionados se presentó el caso de que al buscar la palabra *depression*, se obtenían resultados como *the great depression*, y al buscar *suicide*, se obtenían resultados como *suicide bombers*. Por lo tanto, éstos y otros términos similares se excluyeron.
4. Se analizó la frecuencia de los tuits utilizando el Modelo Autorregresivo Integrado de Media Móvil (ARIMA).

5. Todos los análisis y pronósticos fueron realizados con el software de R versión 3.1.0 utilizando el paquete de pronósticos.

Como podemos apreciar, en este estudio no se utilizó la *API* de twitter, sino que utilizaron un servicio la empresa Crimson Hexagon, la cual está especializada en el análisis de datos en redes sociales, de la cual se pudieron obtener los tuits generados del año 2011 al año 2014.

Si comparamos con las investigaciones anteriormente mencionadas, las cuales utilizaron el *API* de twitter en periodos de tiempo y en el transcurso en el que se desarrolló su investigación, cabe mencionar que la *API* de twitter tiene sus limitantes como por ejemplo, ésta no puede descargar tuits de más de nueve días anteriores a la fecha de consulta. Por lo tanto, el análisis de los datos que se realizó, fue para entender el comportamiento pasado para pronosticar eventos futuros.

En los resultados obtenidos en este estudio que se muestran en la Ilustración 16, se presentaron eventos inesperados, como el polémico suicidio del famoso actor Robin Williams, lo cual aumentó considerablemente en las redes sociales el tema, viéndose reflejado en los resultados del análisis de los datos. El autor menciona de igual manera que se pueden pronosticar aumentos dado eventos previstos, como por ejemplo, el día mundial sobre la prevención del suicidio o el *Bell Let's Talk Day* sobre la prevención de problema de *Bullying*, que de igual manera se vio reflejado en los resultados del análisis de los datos.

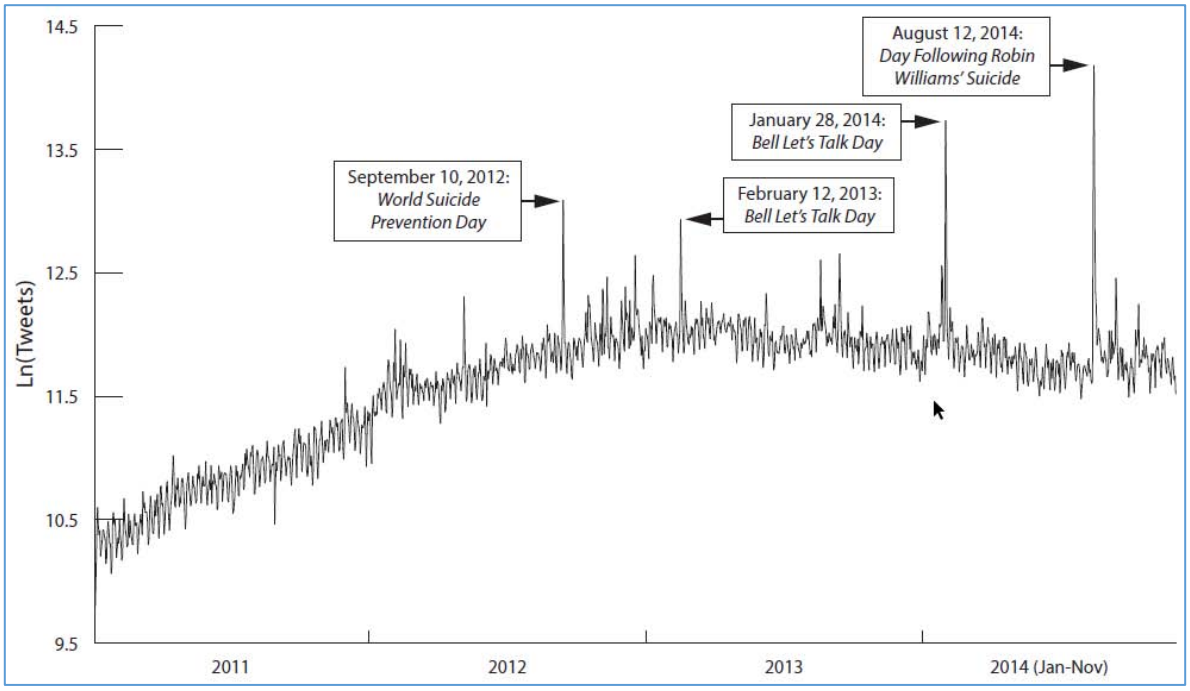


Ilustración 16. Presencia de Menciones acerca del suicidio. (McClellan, Ali, Mutter, Kroutil, & Landwehr, 2016)

En la Ilustración 17, se muestra la medición diaria y el pronóstico del día siguiente sobre los tuits con menciones referentes al suicidio. Aplicando el pronóstico que contiene el programa de R y el modelo de ARIMA, se pudo pronosticar el comportamiento de las menciones, salvo en tres ocasiones en las que sucedieron los eventos mencionados anteriormente.

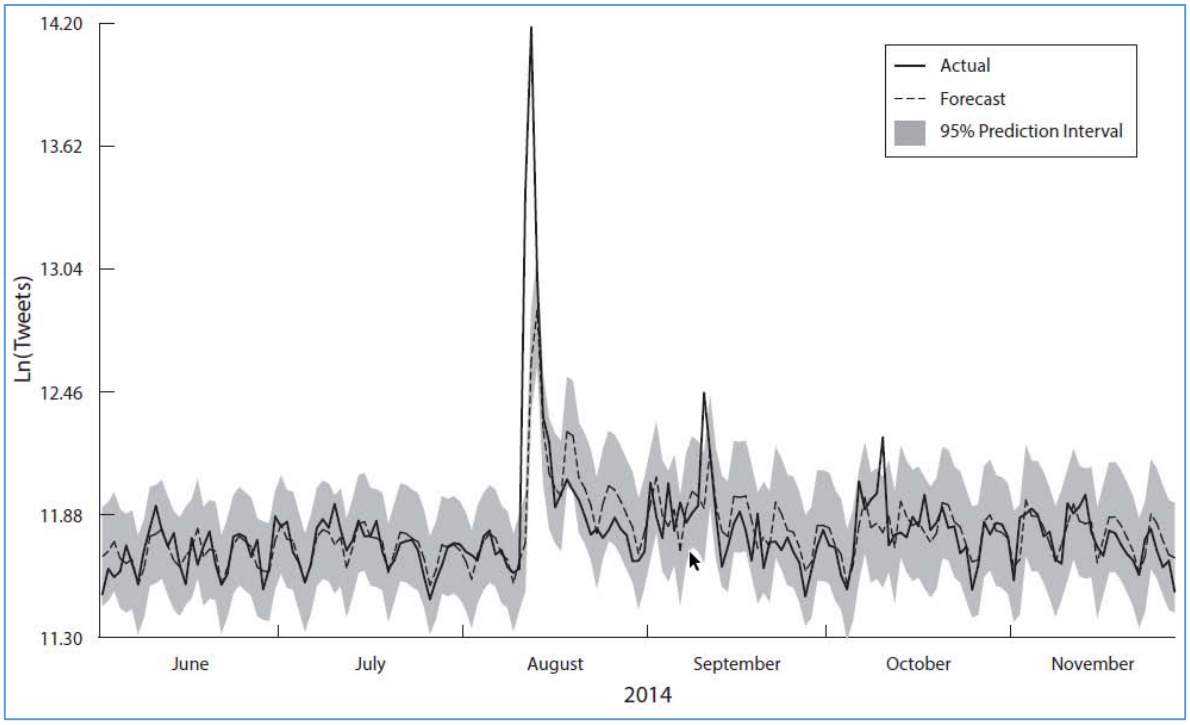


Ilustración 17. Medición real, pronóstico y predicción de las menciones. (McClellan, Ali, Mutter, Kroutil, & Landwehr, 2016)

2.1.4 Análisis de sentimientos de twitter relacionados con la marca utilizando herramientas de ingeniería y la arquitectura dinámica para redes neuronales artificiales.

En la *49th Hawaii International Conference on System Sciences* celebrada en el 2016, se expuso un artículo acerca del análisis de sentimiento en twitter, implementado sobre la marca de Starbucks, en donde se utilizaron técnicas de redes neuronales y aprendizaje automatizado.

Se presentó una investigación respecto al análisis de sentimiento, utilizando aplicaciones para el análisis de sentimiento y una Arquitectura Dinámica para Redes Neuronales Artificiales, llamado DAN2.

En el artículo de la conferencia menciona que para el año 2016, twitter cuenta ya con más de 100 millones de usuarios que generan alrededor de 500 millones de tuits por día. Lo cual evidentemente llama la atención de los investigadores, por lo que se aplicó el análisis de sentimiento en el dominio comercial, utilizando la marca mundialmente reconocida de café Starbucks.

En el artículo expuesto en la conferencia, menciona que el desempeño del análisis de sentimiento es pobre, ya que se alcanza apenas un 70% de certeza o menos en la clasificación de los tuits, ya que twitter por su limitación a 140 caracteres, da relativamente pocos datos para realizar un análisis de sentimiento más exacto. Se ha detectado de igual manera que la mayoría de los tuits no expresan sentimiento negativo o positivo, sino que se encuentran en un estado neutral, por lo que uno de los retos en el análisis de sentimiento, es detectar las infrecuentes ocurrencias de neutralidad. Se detectó de igual manera que los tuits relacionados con marcas comerciales, frecuentemente reflejaban un sentimiento ya sea negativo o positivo. Un dato importante que mencionar en este

estudio, es la importancia que se le ha dado a los *emoticons*, los cuales son capaces de reflejar un estado de ánimo o un sentimiento a las publicaciones. (Zimbra, Ghiassi, & Lee, 2016)

Por lo tanto se entiende que en el análisis de sentimiento se contemplaban tres tipos de tuits (positivo/negativo/neutral), hasta que se propuso la arquitectura dinámica, que hizo posible la nueva clasificación quedando de la siguiente manera: fuertemente positiva, medianamente positiva, neutral, medianamente negativa y fuertemente negativa.

En el artículo publicado por Nakov Preslav (2016), corrobora la escala con cinco clases de tuits, la cual ya está siendo implementada en el mundo corporativo, como por ejemplo en Amazon, TripAdvisor y Yelp, todas ellas utilizan la escala de cinco clases para puntuar sus productos, hoteles y restaurantes respectivamente.

Se obtuvieron 442,443 tuits utilizando la *API* de twitter que contuvieran "@Starbucks" en su contenido, de los meses de agosto a octubre del 2013. Los retuits fueron removidos, quedándose con 254,196 tuits. De manera aleatoria seleccionaron 9,367 tuits que fueron analizados y evaluados manualmente por tres estudiantes. Se utilizaron 5 clases de sentimiento como se muestran en la Tabla 4.

Tabla 4. Descripción de las clases de sentimiento en twitter. (Zimbra, Ghiassi, & Lee, 2016)

Clase de sentimiento	Descripción	Ejemplo
Fuertemente positivo	El autor claramente adora la marca.	The last 2 seasons of my year are defined by @starbucks: Fall = 1st day of Pumpkin Spice Latte. Winter = 1st day of Christmas Blend. #truth
Medianamente positivo	Al autor le gusta la marca.	Somehow we are staying at the only hotel in the vicinity that doesn't have a @Starbucks. This was very poor planning.
Neutral	No se entiende claramente el sentimiento por la marca.	So does someone know how to use the pitcher box of passion tea from @Starbucks? I'm really confused. Or I just can't read.
Medianamente Negativo	Al autor no le gusta la marca.	This mornings #Crossfit workout was hard. But this coffee @Starbucks is harder. #AddSomeMoreH20Yo
Fuertemente Negativo	El autor claramente odia la marca.	@Starbucks I had a horrible experience at your store bad customer service the employee was very rude #Fail

Los tuits seleccionados resultaron en un conjunto de tuits de 5,526, con una distribución en las clases de sentimiento como se muestra en la Tabla 5.

Tabla 5. Distribución de las clases de sentimiento para el conjunto de datos de Starbucks.

Clase de Sentimiento	Tuits
Fuertemente positivo	2885
Medianamente positivo	617
Neutral	414
Medianamente Negativo	783
Fuertemente Negativo	827

Se compararon los resultados de los tuits seleccionados con los siguientes cuatro sistemas de análisis de sentimiento:

- Sentiment140: Utiliza los emoticonos para evaluar el sentimiento y solo devuelve tres clases de sentimiento.
- Repustate: Herramienta de pago, para fines comerciales, con cinco clases de sentimiento.
- SVM: Suport Vector Machine, con cinco clases de sentimiento.
- DAN2: Arquitectura Dinámica para Redes Neuronales Artificiales, con cinco clases de sentimiento.

Y en resumidas cuentas, se tienen los siguientes resultados:

En la Tabla 6, se muestra el porcentaje de precisión en el sentimiento que devuelven las herramientas:

Tabla 6. Porcentaje de precisión en la clasificación del sentimiento.

Clases de Sentimiento	DAN2	SVM	Sentiment140	Repustate
Tres	86.06%	78.33%	39.96%	45.82%
Cinco	85.56%	78.39%	--	34.09%

Por lo que podemos observar que DAN2 y SVM son mejores que Sentiment140 y Repustate.

Capítulo 3. Análisis, diseño y arquitectura

En este capítulo se analizarán los requerimientos y se realizará el diseño, especificando la secuencia de las etapas, la distribución de los datos y el tratamiento que se le dará a los mismos, se detallará la arquitectura que se utilizará y la descripción de los elementos que la componen, basándose en el ciclo de vida del *Big Data* de Thomas Erl descrito en la sección 1.3.2.

En la Tabla 7. Se muestran las etapas del ciclo de vida del *Big Data* de Thomas Erl, que se mencionó anteriormente.

Tabla 7. Etapas del Big Data.

Ciclo de vida del *Big Data*.
1. Evaluación del caso de negocio.
2. Identificación de los datos.
3. Adquisición de los datos.
4. Extracción de los datos.
5. Validación de los datos y limpieza.
6. Agregación de los datos y representación.
7. Análisis de los datos.
8. Visualización de los datos.
9. Usos de los resultados del análisis.

El ciclo de vida propuesto por Min Chen descrito en la sección 1.3.1, no cubre con el suficiente detalle como el propuesto por Thomas Erl mencionado anteriormente, ya que Min Chen en su artículo, lo separa en tres grandes etapas muy generales, y Thomas Erl los separa en nueve etapas más específicas con objetivos más claros para realizar un proyecto de *Big Data*. La Administración del Ciclo de Vida del Producto descrito en la sección 1.4, está muy relacionado con la producción de artículos tangibles, relacionando conceptos de procesos de producción y calidad industrial, con un proyecto de *Big Data*, por lo que al querer implementar el proyecto bajo los conceptos que propone el PLM, puede confundir el contexto de la implementación de un proyecto de *Big Data*.

Como se vio en la sección 1.1, el objetivo de un análisis de *Big Data*, estudia y analiza los datos generados por los procesos de trabajo de una empresa, ya que utilizando los resultados se puedan detectar problemas, para que posteriormente se puedan modificar y añadir procesos para mejorar y planificar nuevas estrategias, y fortalecer debilidades detectadas.

3.1 Análisis y diseño del ciclo de vida

El proyecto se basará en el ciclo de vida del *Big Data* propuesto por Thomas Erl en la sección 1.3.2, y no se considerará la novena etapa sobre el uso de los resultados del análisis como tal, ya que no hay usuarios de negocio, quienes son los que toman decisiones con base en los resultados obtenidos del análisis.

Se realizará el estudio de *Big Data* utilizando twitter como fuente de datos, para ver que relación o similitud hay en la presencia y apariciones de drogras en menciones de twitter, con el resultado de

la encuesta nacional del consumo de drogas en los Estado Unidos que se publicó en el 2015. El objetivo de este estudio, es poder comparar la presencia de temas relacionados con las drogas en twitter, contra los de la encuesta realizada, detectar su presencia por zona geográfica y el tiempo, y realizar un análisis de sentimiento sobre los tuits que se vayan a procesar.

3.1.1 Evaluación del caso de negocio

El estudio se realizará en el tema de la salud pública, el cual estará enfocado en el consumo de drogas y la aparición de las mismas en twitter. Se aplicará un análisis de *Big Data*, para detectar en las 10 ciudades más pobladas de la unión americana, la presencia de palabras y términos relacionados con las 10 drogas más consumidas.

De acuerdo al resultado de la encuesta realizada en el 2015, por *la National Survey on Drug Use and Health* (NSDUH), se sabe que hay 27.1 millones de consumidores de drogas mayores de 12 años en la unión americana, y se concentran en diez conjuntos principales. (Substance Abuse and Mental Health Services Administration, 2015)

En la Ilustración 18, podemos observar que está encabezando la lista el consumo de la mariguana.

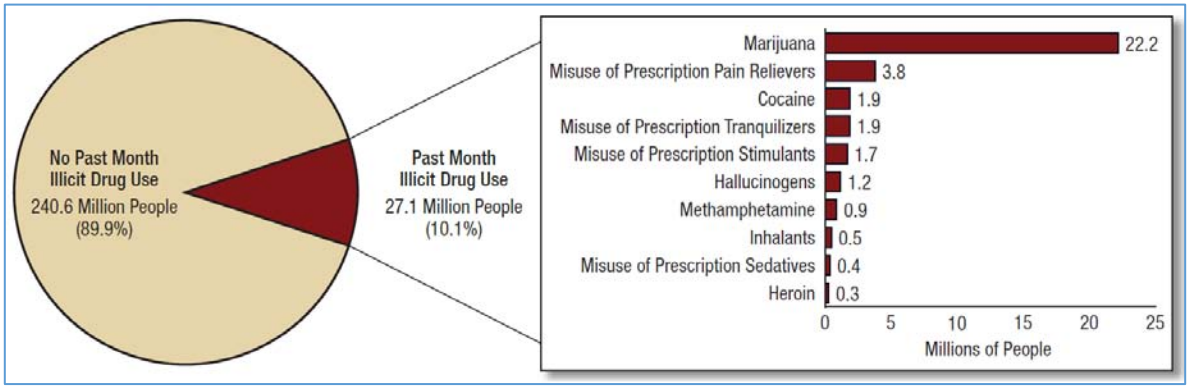


Ilustración 18. Consumidores mayores de 12 años en el 2015. (Substance Abuse and Mental Health Services Administration, 2015, pág. 7)

Se buscará encontrar una relación entre las apariciones de las menciones de las drogas en las redes sociales, comparándolas con los resultados de la encuesta, y así poder crear un modelo de monitoreo, cubriendo el requisito de velocidad al requerirse un análisis en tiempo real, y variedad, al comparar los datos de twitter contra los de las encuestas, para determinar: a) tendencias en el consumo de drogas en la unión americana, b) sentimiento de las menciones en twitter, y c) detectar por ciudad la menciones recopiladas. Y por último, volumen, el cual se cubre al tener un flujo de datos masivos constante provisto por twitter.

Se pretende contestar las siguientes preguntas, las cuales tienen un enfoque exploratorio, ya que no se sabe por ejemplo, cual es el sentimiento de los usuarios que realizan las menciones, a diferencia del confirmatorio, en el cual se puede suponer que el consumo de drogas se eleva los fines de semana, similar al trabajo de investigación sobre el consumo de alcohol de la sección 2.1.1:

1. De acuerdo a los resultados de las encuestas, ¿existe una presencia en twitter similar a las 10 drogas más consumidas reportadas por la encuesta de la NSDUH?
2. ¿Se podrán detectar picos en el consumo de drogas monitoreando twitter en tiempo real?

3. ¿Se podrá detectar si los usuarios consumidores son adictos?
4. ¿Se podrán detectar distribuidores de drogas?
5. ¿Se podrá visualizar la presencia de drogas por día en determinadas áreas geográficas?
6. ¿Se podrá determinar en qué área está teniendo predominancia una droga en tiempo real?
7. En las menciones donde aparecen drogas, ¿Se podrán identificar los grupos de drogas más comunes?
8. Aplicando análisis de sentimiento, ¿Se podrá obtener el resultado de la agrupación, aplicado solamente a los tuits con menciones de drogas en general y por ciudad?
9. De los 10 usuarios con el mayor número de menciones relacionadas con drogas, ¿Se podrá saber cuántas de sus menciones en total contienen alguna droga?

Para realizar el proyecto, se requiere como inversión, utilizar equipo de cómputo de alta gama, con memoria RAM suficiente, ya que se utilizará el marco de trabajo libre de Hortonworks que demanda mucha memoria RAM. Se utilizará el servicio de cómputo en la nube que provee Amazon en su servicio de instancias virtuales EC2, para ejecutar el consumo del flujo de datos de manera ininterrumpida.

3.1.2 Identificación de los datos

Se realizará un análisis sobre las publicaciones que publican los usuarios en twitter. Esta red social como se mencionó en el capítulo 1, cuenta con el 24% de usuario de redes sociales. Por lo que nos da una muestra bastante aceptable de personas activas y que publican sus puntos de vista, ideas, estados de ánimo, etc.

El análisis se efectuará con datos provenientes de twitter, ya que es la única red social que provee una *API* para consultar mediante servicios web, las publicaciones realizadas por los usuarios, con una gran variedad de filtros para obtener resultados específicos.

Por otra parte, twitter es una red social que no censura temas como el consumo de drogas, permitiéndonos obtener datos que no podríamos obtener de otra red social. No se utilizó Facebook por que no cuenta con ningún método libre, similar al de twitter, para descargar publicaciones ya que Facebook se reserva estas herramientas para clientes particulares.

3.1.3 Adquisición de los datos y filtrado

Los datos disponibles de twitter, haciendo uso del *API* que provee, se utilizarán para realizar las consultas a todos aquellos tuits, filtrando por idioma inglés y que sean publicados sobre regiones previamente definidas, ubicándolas por latitud y longitud de las ciudades sobre las que se hará el análisis.

El *API* de twitter responde hasta los últimos 100 tuits publicados, por lo cual, se deberá de validar que el tuit sea nuevo, esto basándonos en el identificador único con el que cuentan. Cada tuit trae la información general del usuario que realizó la publicación, por lo que almacenaremos todos los usuarios de quienes se registren publicaciones, y los *hashtags* que contienen los tuits.

Los parámetros utilizados y que recibe el *API* de twitter para realizar consultas, son los siguientes:

Parámetro	Requerido	Descripción	Ejemplo
q	Si	Codificado en UTF-8 y tipo URL, máximo 500 caracteres, incluyendo operadores lógicos. Puede ser limitada la respuesta de acuerdo a la complejidad de la consulta.	*UNAM* *Posgrado* *PCIC* *Big Data*
geocode	opcional	Especifica que regrese los tuits de los usuarios en determinada latitude y longitude, adicionalmente el radio en millas o kilómetros del punto seleccionado.	*37.78 -122.39 1mi*
lang	opcional	Consulta solamente los tuits realizados en determinado idioma, basándose en el código de idiomas ISO 639-1.	*eu*
result_type	opcional	Especifica que tipo de tuits queremos recibir, teniendo los siguientes tres tipos: • *mixed* : Mezcla en su respuesta los más populares en el momento y los más recientes. • *recent* : Regresa los tuits más recientes. • *popular* : Regresa los tuis más populares.	*mixed* *recent* *popular*
count	opcional	Especifica el número de tuits por consulta. Limitado a máximo 100 y por default 15.	*100*

Un ejemplo de la consulta, utilizando el método REST quedaría de la siguiente manera:

```
GET
https://API.twitter.com/1.1/search/tweets.json?q=%23freebandnames&
result_type=mixed&count=4
```

Y la respuesta en el formato JSON es de la siguiente manera:

```
{"statuses": [
    { "coordinates": null,
      "favorited": false,
      "truncated": false,
      "created_at": "Mon Sep 24 03:35:21 +0000 2012",
      "id_str": "250075927172759552",
      "entities": {
        "urls": [],
        "hashtags ": [{
             "text": "freebandnames",
           "indices": [20,34]}],
        "user_mentions": []},
      "in_reply_to_user_id_str": null,
```

```
"contributors": null,
"text": "Aggressive Ponytail #freebandnames",
"metadata": {
  "iso_language_code": "en",
  "result_type": "recent"
},
"retweet_count": 0,
"in_reply_to_status_id_str": null,
"id": 250075927172759552,
"geo": null,
"retweeted": false,
"in_reply_to_user_id": null,
"place": null,
"user": {
  "profile_sidebar_fill_color": "DDEEF6",
  "profile_sidebar_border_color": "C0DEED",
  "profile_background_tile": false,
  "name": "Sean Cummings",
  "profile_image_url": "http://a0.twimg.com/normal.jpeg",
  "created_at": "Mon Apr 26 06:01:55 +0000 2010",
  "location": "LA, CA",
  "follow_request_sent": null,
  "profile_link_color": "0084B4",
  "is_translator": false,
  "id_str": "137238150",
  "entities": {
    "url": {
      "urls": [{
          "expanded_url": null,
          "url": "",
          "indices": [0,0]}]},
    "description": {
      "urls": [ ]}},
  "default_profile": true,
  "contributors_enabled": false,
  "favourites_count": 0,
  "url": null,
  "profile_image_url_https":"normal.jpeg",
  "utc_offset": -28800,
  "id": 137238150,
  "profile_use_background_image": true,
  "listed_count": 2,
  "profile_text_color": "333333",
  "lang": "en",
  "followers_count": 70,
  "protected": false,
  "notifications": null,
```

```
        "profile_background_image_url_https": "bg.png",
        "profile_background_color": "C0DEED",
        "verified": false,
        "geo_enabled": true,
        "time_zone": "Pacific Time (US & Canada)",
        "description": "Born 330 Live 310",
        "default_profile_image": false,
        "profile_background_image_url":
"http://a0.twimg.com/images/themes/theme1/bg.png",
        "statuses_count": 579,
        "friends_count": 110,
        "following": null,
        "show_all_inline_media": false,
        "screen_name": "sean_cummings"
      },
      "in_reply_to_screen_name": null,
      "source": "twitter for Mac",
      "in_reply_to_status_id": null
    }
}
```


Por lo tanto, los datos que nos serán útiles son los siguientes:

a) Tuit:
   i. created_at: La fecha y hora de publicación del tuit por el usuario.
   ii. text: Contenido textual de la publicación.
   iii. id: Identificador numérico único de cada tuit.
   iv. Retweeted: Nos devuelve True si el tuit está siendo utilizado por un usuario como referencia en un tuit nuevo, con la intención de compartir la publicación de otro usuario de twitter.

b) Usuario:
   i. name: Nombre del usuario, el cual no precisamente es una persona, puede ser una institución, empresa, dependencia de gobierno, etc.
   ii. location: El nombre de la localización donde reside el usuario, este campo abierto que se registra sin el uso de algún estándar.
   iii. id: Identificador único de cada usuario.
   iv. followers_count: Número de seguidores de la cuenta del usuario.
   v. description: Descripción que el usuario registra de sí mismo.
   vi. friends_count: Número de personas a las que sigue el usuario.
   vii. screen_name: Nombre del usuario público con el cual se hace referencia al mismo, al usar el carácter de arroba, por ejemplo @unam_mx.

c) Hashtag:
   i. text: Palabra o palabras, usualmente utilizando un estilo *camel case* para apoyar una idea o frase, precedidas por el carácter hashtag, por ejemplo #UnamGlobal.

Los parámetros de consulta que se utilizarán, serán los siguientes.

Parámetro	Valor
q	*Sí utilizar.*
geocode	*Asignado con base en el catálogo de ciudades por latitud y longitud de EUA.*
lang	*Únicamente los que estén en idioma Inglés "eu".*
result_type	*Solamente los recientes "recent "*
count	*De 1-100 tuits más recientes.*

El parámetro q, el cual sirve para buscar tuits que contengan textos específicos, no se asigna, ya que queremos obtener todos los tuits sin importar el contenido.

El diagrama entidad-relación de los datos útiles se muestra en la Ilustración 19:

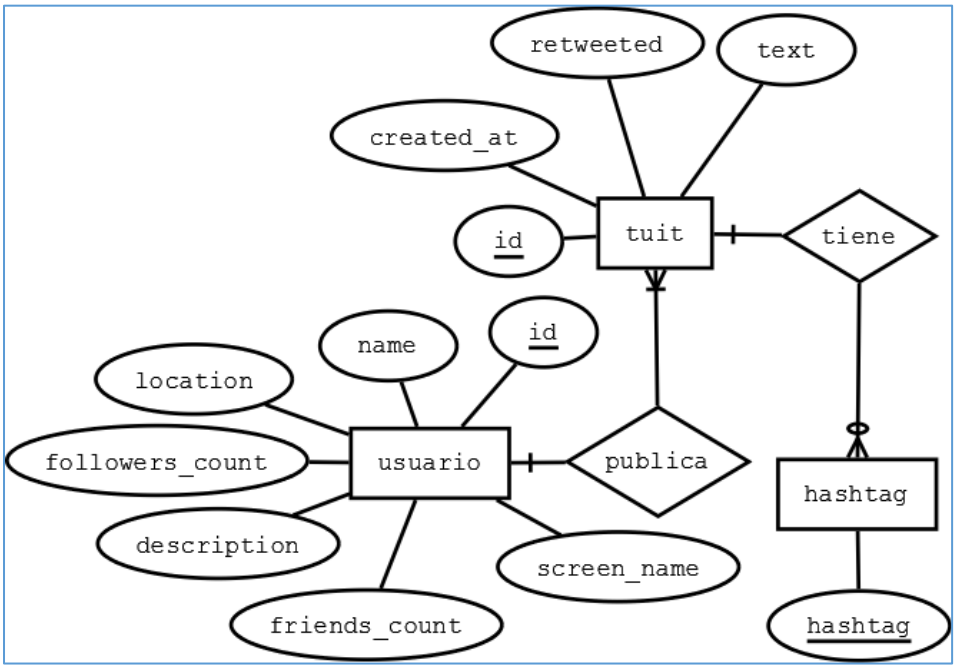


Ilustración 19. Diagrama E-R de los datos útiles que provee la API de twitter.

Al interpretar el diagrama con la descripción de los datos obtenidos por la *API*, tenemos que un usuario, el cual tiene un identificador único, un nombre de usuario, el nombre de la ubicación en la que reside, el número de seguidores, el número de seguidos, la descripción de sí mismo y un nombre de usuario público, el cual puede realizar publicaciones de tuits. Un tuit tiene un identificador único, una fecha de creación, el contenido en texto de la publicación y un identificador para saber cuándo un tuit es un retuit. Y por último, tenemos identificado que un tuit puede tener *hashtags*.

Una vez identificadas las entidades y sus atributos que serán la base de nuestro análisis, se requerirá añadir nuevas entidades y atributos para poder realizar el análisis de los datos.

La entidad de ciudad será añadida, ya que los tuits se extraerán de ciudades predeterminadas, por lo que un tuit tiene una ciudad. Se agregará una entidad que contenga los grupos de palabras que

serán buscados en el texto del tuit, por lo tanto, un grupo de palabras agrupa varias palabras. Se agregará otra entidad que contenga las palabras de la Tabla 8 del grupo de palabras, y una última entidad que contenga la relación entre el tuit y la palabra a buscar en el texto, entendiéndose que un tuit puede contener alguna palabra del grupo de palabras. Se añadirá el atributo sentimiento, resultante del análisis de sentimiento que se tiene previsto, en la entidad tuit.

Tabla 8. Palabras que se buscarán en el contenido de los tuits.

Palabras		
beer	angel dust	crank
alcohol	heroin	methadrine
cerveza	*API*ates	aspirin
vodka	snort	morphine
tequila	weed	opioids
whisky	marijuana	fentanyl
whiski	kush	hydrocodone
whyski	cannabis	methadone
whiskey	mariguana	opiate
cocaine	canabis	opium
coke	ganja	barbiturates
crack	bhang	amphetamine
happy dust	hashish	big h
nose candy	mojo	analeptic
stardust	Methamphe-tamine	tabaco
white horse	meth	cigar
white lady		

La representación de las nuevas entidades descritas se modela en el diagrama entidad-relación que se muestra en la Ilustración 20.

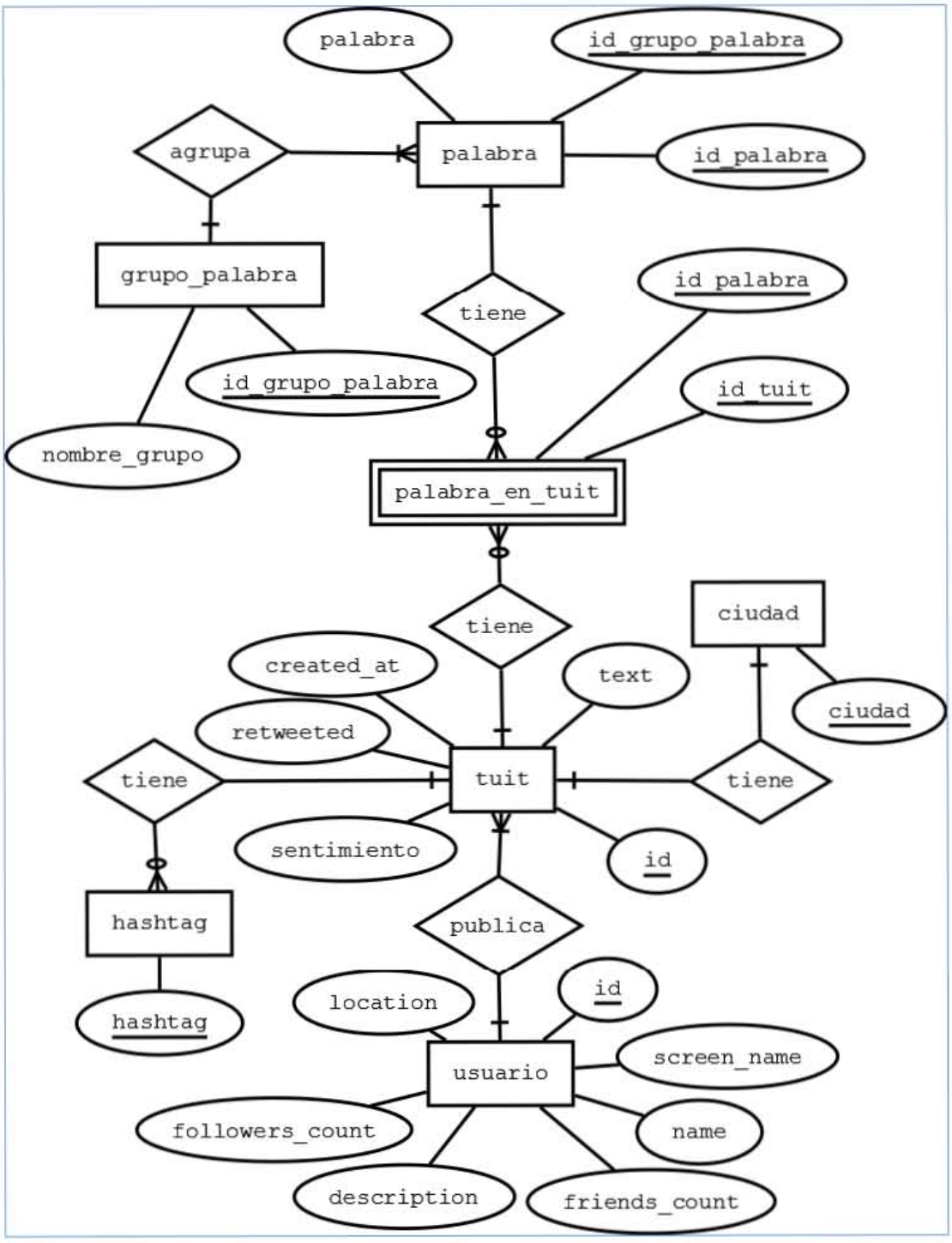


Ilustración 20. Diagrama E-R con entidades añadidas.

Para acercarnos al modelo físico de datos, se tiene el siguiente modelo relacional, al cual se le agregan atributos para poder tener una mayor integridad referencial como se muestra en la Ilustración 21.

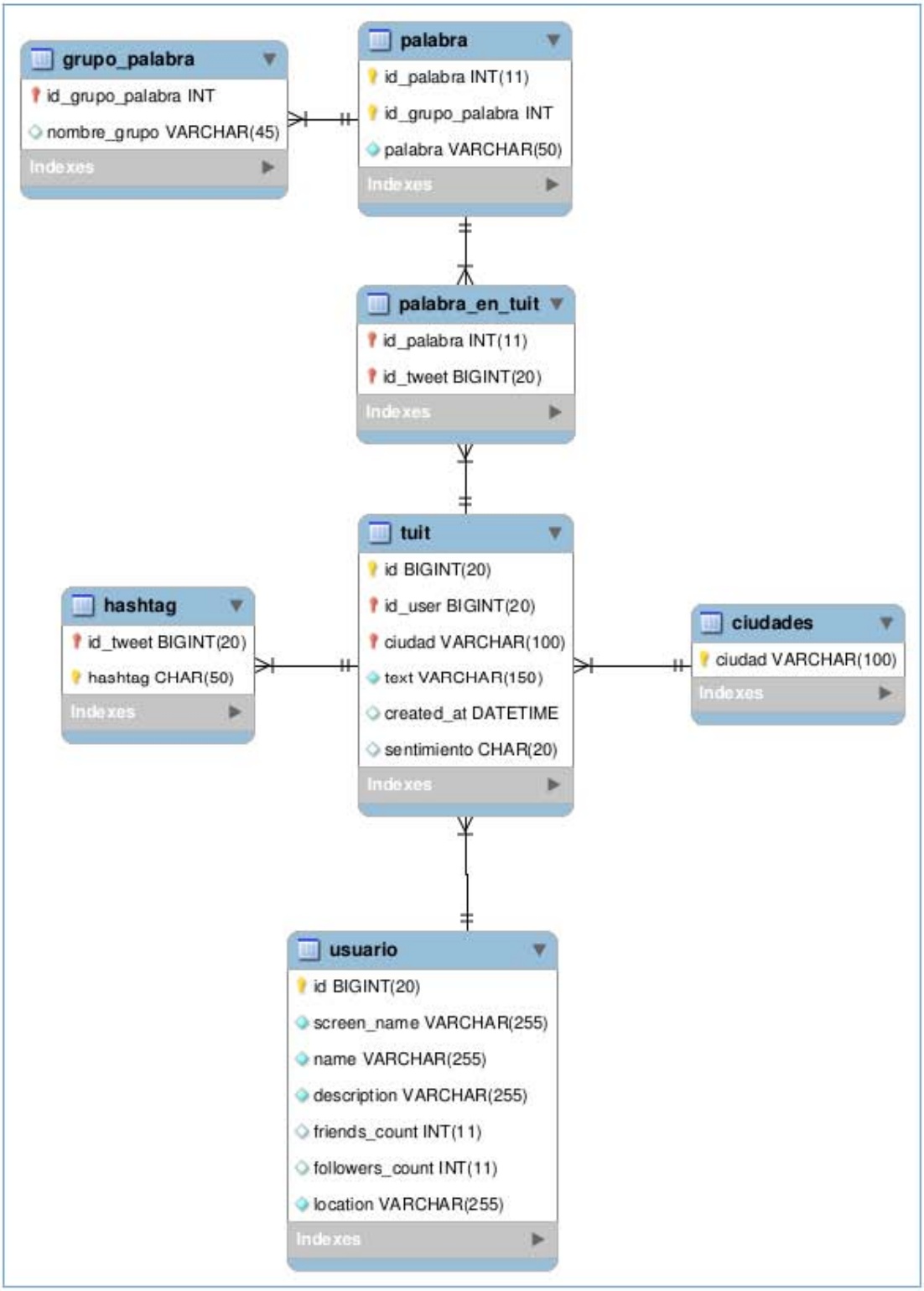


Ilustración 21. Modelo relacional de los datos provenientes de twitter.

El modelo refleja los siguientes cambios importantes:

1. Como podemos observar, a la entidad tuit se le agregó el id_user para identificar el usuario que publica el tuit y la ciudad desde donde se hizo.
2. El atributo de retuit no se considera en este diagrama, ya que es un dato que no se almacenará, solamente será utilizado para saber si es retuit o no al momento de consultar los tuit, y de ser True el valor de este atributo, no se almacenará.
3. La entidad hashtag se le agregó el atributo de id_tweet, y junto con el atributo hashtag, se crea una llave compuesta.

La secuencia para la adquisición de los datos, en donde nuestro programa deberá conectarse a la *API* de twitter para obtener los tuits, se define de la siguiente manera:

1. Tendremos un actor, ya sea el usuario o un proceso automático, el cual ejecutará el programa que inicia la consulta.
2. El programa configura los parámetros de búsqueda de la *API* de twitter.
3. El *API* realiza la consulta de tuits.
4. El *API* recibe los tuits resultantes de la búsqueda.
5. Los tuits recibidos son analizados para obtener el sentimiento de cada uno.
6. Los tuits y el resultado del análisis de sentimiento son almacenados.
7. Se cierra la consulta.

El diagrama de secuencia se muestra en la Ilustración 22.

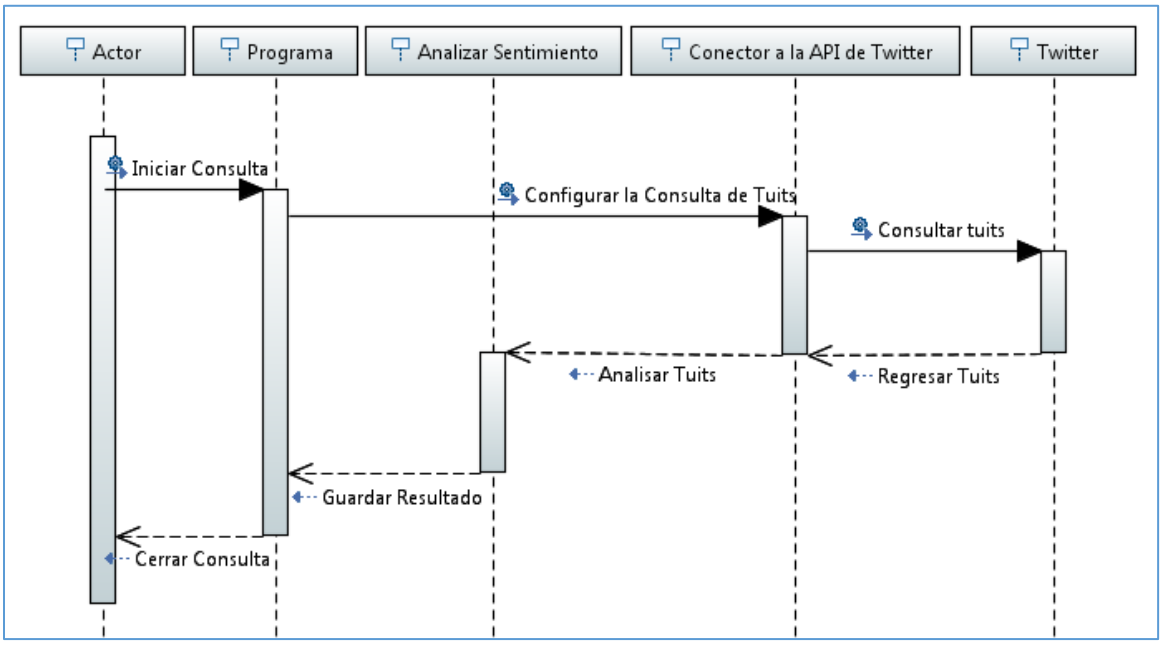


Ilustración 22. Diagrama de secuencia de la adquisición de datos.

3.1.4 Extracción de los datos.

La extracción de los datos de nuestras fuentes de almacenamiento es necesaria, ya que se utilizarán herramientas especializadas para *Big Data*, lo que implica extraer los datos almacenados de los contenedores de datos principales, los cuales suelen estar en uso constante y nos limita el acceso a los mismos. Por lo tanto, se deberá realizar una extracción de los datos al ambiente de trabajo con capacidades analíticas de *Big Data*.

De los contenedores de datos, se deberá revisar la estructura para seleccionar cuales nos serán útiles para realizar el análisis de *Big Data*, pre-procesarlos previamente para ser almacenados en el marco de trabajo para el análisis de los datos.

La secuencia para extraer los datos se define en los siguientes pasos:

1. El actor, ya sea un programa o un usuario, efectuará el proceso de sincronizar los datos.
2. Se deberá tener el contenedor de datos en el marco de trabajo de *Big Data* configurado, para que este a su vez consulte en sí mismo el último registro añadido, para poderlo comparar con el contenedor de datos principal, y esto para que únicamente los registros nuevos se añada al de *Big Data*.
3. Finaliza la sincronización.

En el siguiente diagrama de secuencia de la Ilustración 23, se muestra como deberá de comportarse el programa que extraerá los datos.

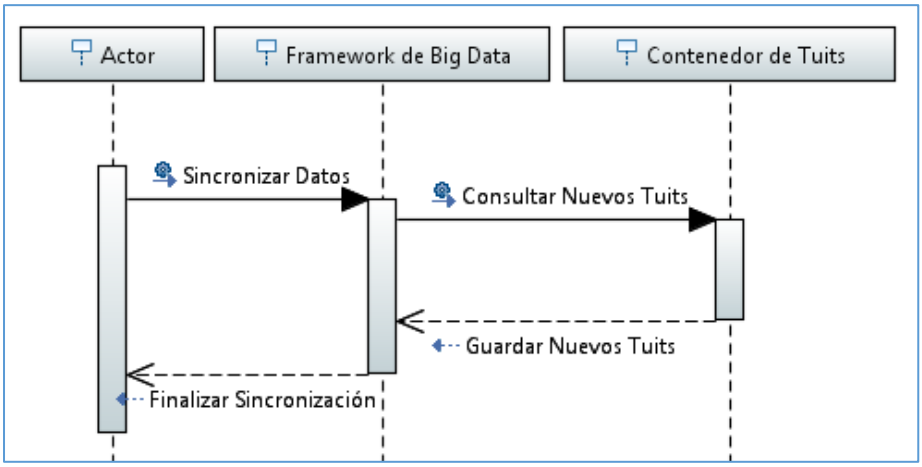


Ilustración 23. Diagrama de secuencia de la extracción de datos.

3.1.5 Validación de los datos y limpieza

El proceso de validación se contemplará desde la adquisición de los datos provenientes de la *API* de twitter. Las reglas para considerar que los datos obtenidos sean validados y estén limpios de cualquier valor corrupto o repetido, son las siguientes:

1. Se deberá validar el tuit, basándose en el identificador único, que éste se agregue al contenedor de datos una sola vez con sus respectivos *hashtags* y usuario.
2. Se deberá validar el usuario, basándose en el identificador único, que se agregue al contenedor de datos una sola vez.
3. Se deberá limpiar el texto de la publicación del tuit, eliminando caracteres ilegibles.

3.1.6 Agregación de los datos y representación

Dentro del marco de trabajo para el análisis de *Big Data*, se definen los procesos de agregación de datos, para realizar las consultas que permitan contestar las preguntas definidas en la evaluación del caso de negocio.

Las entidades están relacionadas con los identificadores, como podemos ver en la Ilustración 21, donde está identificado con el id_usuario que proviene de la entidad usuario y su ciudad con la entidad del mismo nombre. Esto permitirá agregar y representar los datos en nuevas vistas para su análisis, como se verá más adelante en la implementación de la visualización de los datos.

3.1.7 Análisis de los datos

Una vez teniendo todos los datos concentrados en el marco de trabajo de *Big Data*, esta etapa se concentra en encontrar las respuestas a las preguntas del caso de negocio, y de ser posible, encontrar patrones en los resultados obtenidos, por lo que se realizarán consultas utilizando agrupaciones de los datos contenidos en modelo relacional de la Ilustración 21. Para obtener las respuestas de la preguntas, se plantea el análisis siguiente por cada pregunta:

1. De acuerdo a los resultados de las encuestas, ¿Existe una presencia en twitter similar a las 10 drogas más consumidas reportadas por la encuesta de la NSDUH?
    - La consulta se deberá realizar, en base a la relación de la tabla que contiene los tuits, con la tabla que contiene el identificador del tuit con el identificador de la palabra relacionada con una droga, se deberá de realizar la agrupación sobre la palabra contenida en el catálogo de palabras en el tuit, para obtener el número de menciones por cada una, para después compararla con la reportada por la NSDUH.
2. ¿Se podrán detectar picos en el consumo de drogas monitoreando las redes sociales en tiempo real?
    - Se deberá de sincronizar el marco de trabajo con la fuente de datos, para realizar la consulta de los tuits con menciones sobre drogas más recientes, para poder visualizar en tiempo real el comportamiento de las menciones.
3. ¿Se podrá detectar si los usuarios consumidores son adictos?
    - Relacionando la tabla de tuits, con la tabla de usuarios y, la tabla que contiene la relación de tuits con el catálogo de palabras a buscar, se deberá realizar una consulta sobre el número de menciones que realizan, para poder conocer los usuarios con el mayor número de menciones de drogas, y así poder posteriormente investigar manualmente sus perfiles en twitter para determinar si son adictos o no.
4. ¿Se podrán detectar distribuidores de drogas?
    - Similar a la pregunta anterior, se deberá de identificar si el usuario tiene comportamiento de consumidor o de vendedor de manera manual.
5. ¿Se podrá visualizar la presencia de drogas por día en determinadas áreas geográficas?
    - Relacionando la tabla de tuits y las palabras contenidas en los tuits, se deberá agrupar por ciudad y por fecha, para poder tener el comportamiento de las menciones por ciudad y por fecha.
6. ¿Se podrá determinar en qué área está teniendo predominancia una droga en tiempo real?
    - Similar al análisis para la pregunta 2, pero en este caso, se obtiene la respuesta agrupando por la ciudad.

7. En las menciones donde aparecen drogas, ¿Se podrán identificar los grupos de drogas más comunes?
    - Realizando una serie de agrupaciones y concatenaciones, de las palabras relacionadas con drogas contenidas en el tuit, se podrán obtener los grupos de drogas que aparecen en las menciones.
8. Aplicando análisis de sentimiento, ¿Se podrá obtener el resultado de la agrupación, aplicado solamente a los tuits con menciones de drogas en general y por ciudad?
    - La tabla que contiene los tuits, tiene una columna que contiene el sentimiento resultante del análisis que se realizará por la *API* de Stanford, el cual se puede agrupar por ciudad y por fecha, una vez realizadas las relaciones entre las tablas que contienen la palabra del tuit.
9. De los 10 usuarios con el mayor número de menciones relacionadas con drogas, ¿Se podrá saber cuántas de sus menciones en total contienen drogas y cuantas no?
    - Para obtener este resultado, se consulta el número de menciones totales por usuario, contra el número de menciones relacionadas con drogas. De esta manera se sabrá, si un usuario promueve las drogas con base en sus menciones recurrentes sobre el tema.

3.1.8 Visualización de los datos

De los resultados obtenidos, se definirán las mejores formas para visualizar los resultados, ya que se prevé que puedan ser entre 30 y 50 gráficas que muestren todos los resultados del análisis, por lo que se diseñará una distribución de las gráficas, agrupándolas por tema, pregunta, ciudad, etc.

La secuencia de generar los reportes se define de la siguiente manera:

1. El actor, sea un usuario o un proceso automático, solicita la generación de reportes la cual da paso a actualizar los resultados del análisis.
2. El proceso de actualizar resultados del análisis, implica consultar y generar las gráficas con los datos más actualizados.
3. Al terminar de actualizar los resultados, estos se verán reflejados inmediatamente en el panel de análisis.

En el siguiente diagrama de secuencia de la Ilustración 24, se muestra cómo deberá de comportarse el programa que visualiza los datos.

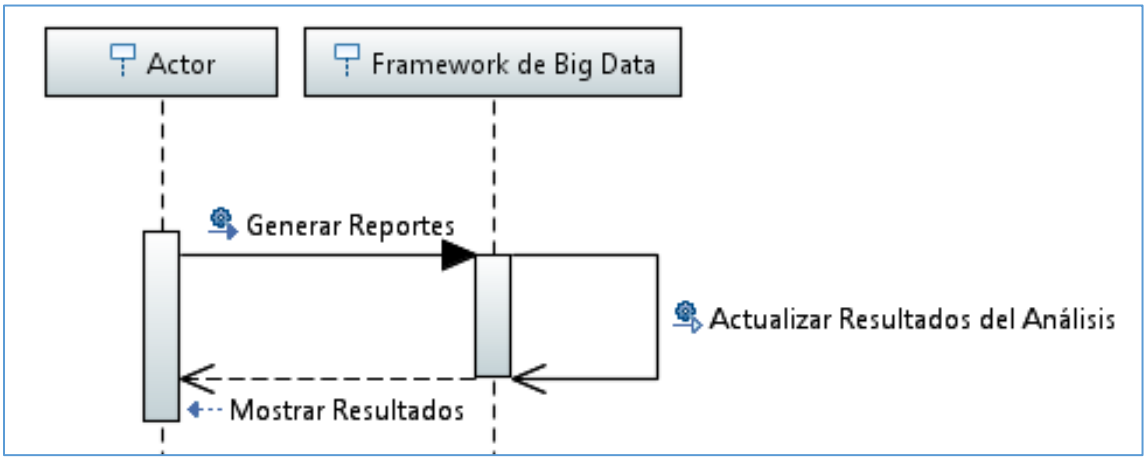


Ilustración 24. Diagrama de secuencia de la visualización de los datos.

3.1.9 Usos de los resultados del análisis

A pesar de no tener un usuario de negocio en este proyecto, los resultados pueden servir para conocer la presencia por zonas geográficas, lo cual las encuestas no lo dan con precisión, permitiendo de esta manera, implementar programas de prevención de consumo de drogas, enfocándose en las de mayor presencia. De igual manera, al conocer los picos en los días de la semana y más aún, en las horas del día, se pueden implementar programas que activen vigilancia adicional cuando se detecten estos picos, para ayudar a por ejemplo, a detectar conductores de automóviles con efectos de alguna droga.

Por otra parte, aprovechando las bondades del análisis de sentimiento, conocer el sentir de las personas de una población respecto a una droga, puede servir para promover o analizar los temas de la legalización que está teniendo auge en la Unión Americana.

3.2 Arquitectura

La arquitectura estará compuesta por 3 nodos principales. El primero de ellos lo identificamos como twitter, que es nuestra fuente de datos, y mediante el *API* realizaremos consultas desde el segundo nodo, identificado como servidor de trabajo, el cual es el que se encuentra en operación constante para obtener los tuits, analizar su sentimiento, validar los datos y almacenarlos en la base de datos. Nuestro tercer nodo se compone de un equipo de cómputo, que contendrá el ambiente de trabajo de análisis de *Big Data*, en donde se contendrán los datos extraídos para ser analizados, y las interfaces necesarias para observar los resultados del análisis en sus respectivos gráficos.

En el diagrama de despliegue de la Ilustración 25, se describe la arquitectura física de los componentes de software con los del hardware.

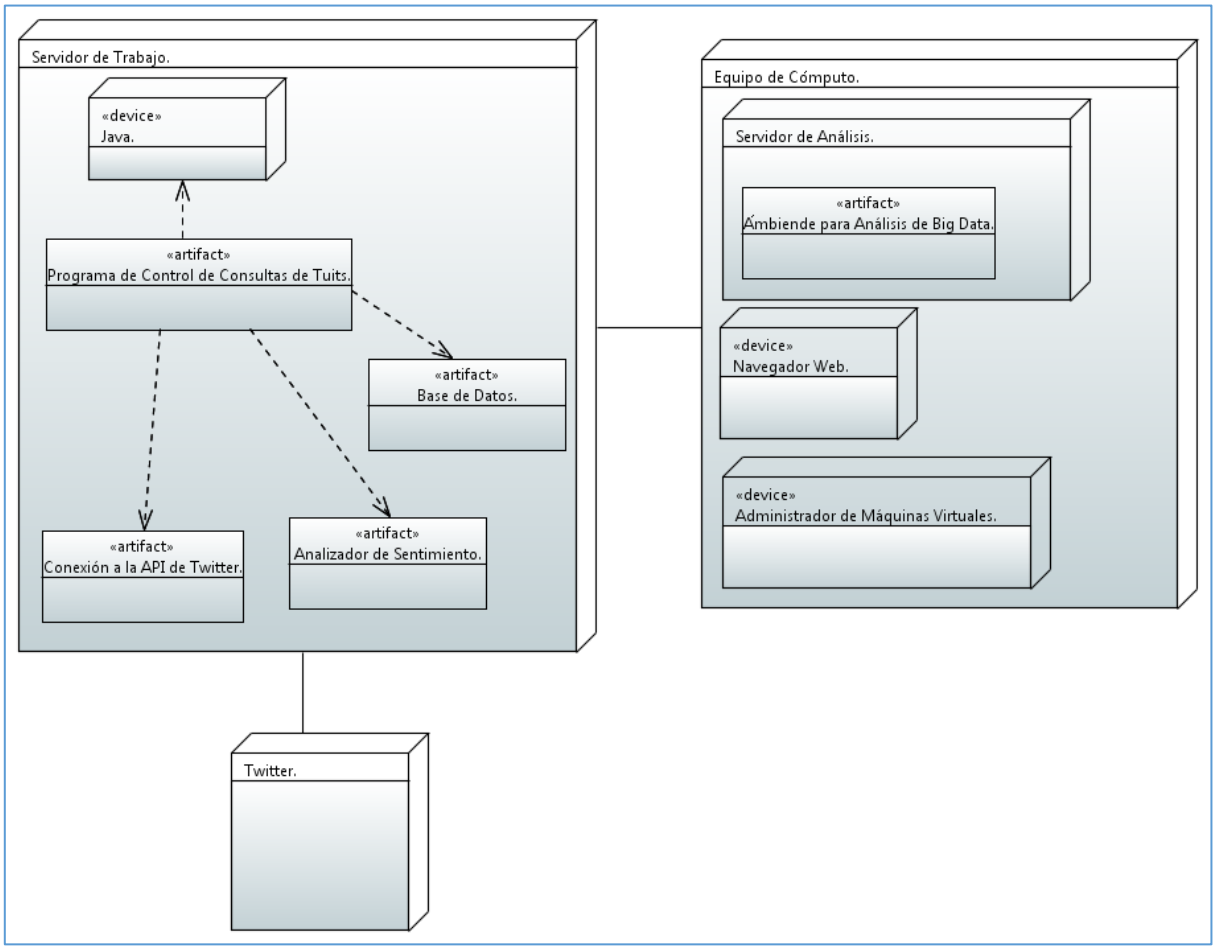


Ilustración 25. Diagrama de despliegue de la arquitectura.

Capítulo 4. Implementación del ciclo de vida del *Big Data*

En este capítulo se abordarán tres de las nueve etapas de la implementación del ciclo de vida de Thomas Erl, iniciando por la Adquisición de los Datos y Filtrado, en donde se verá cómo se realiza la conexión a la *API* de twitter, el procesamiento de los tuits y el análisis de sentimiento, posteriormente, se realizará la Validación de los Datos y Almacenamiento, en donde se verá cómo serán almacenados los tuits, qué infraestructura será utilizada y cómo será la secuencia de ejecución del programa. Una vez terminada dicha etapa, se procederá a pasar a la etapa de Extracción de los Datos, en donde se verá cómo se importarán los datos al marco de trabajo del *Big Data* y la manera en que serán sincronizados los datos. Por último, se explicará cómo se utilizan las herramientas de *Hive*, *Pig Latin* y Zeppelin para el análisis de datos, que provee Hortonworks.

4.1 Adquisición de los datos y filtrado

En la sección 3.1.3 se habló de utilizar el *API* de twitter para obtener los tuits. Para realizar la consulta, se utilizarán las siguientes herramientas:

1. Biblioteca en java Twitter4J, la cual provee los métodos necesarios para realizar consultas y operaciones con el *API* de twitter.
2. IDE de desarrollo Eclipse neon.
3. Una cuenta de twitter activa.

twitter cuenta con el sitio web *dev.twitter.com* con la documentación necesaria para utilizar su *API*, con el fin de facilitar la integración de terceros con sus servicios. Con dicha *API* se pueden hacer actividades de lectura y escritura de datos sobre twitter. En la documentación, se encuentran recomendaciones de uso de bibliotecas en diferentes lenguajes para ser utilizadas, entre ellas se encuentra Twitter4J que se utilizará para la adquisición de los datos.

4.1.1 Biblioteca Twitter4J

Twitter4J es una biblioteca no oficial, con la cual se puede integrar fácilmente aplicaciones escritas en java, a los servicios de twitter. Trabaja con plataformas java superiores a la versión 1.5., es compatible 100% con la *API* 1.1 de twitter, y funciona con Windows y Linux con soporte de Java.

Para poder realizar una conexión con el *API*, se debe de registrar una cuenta de desarrollo en donde se dará de alta la aplicación, para que así sean asignadas claves de acceso necesarias para consumir los servicios, como se muestra en la Ilustración 26.

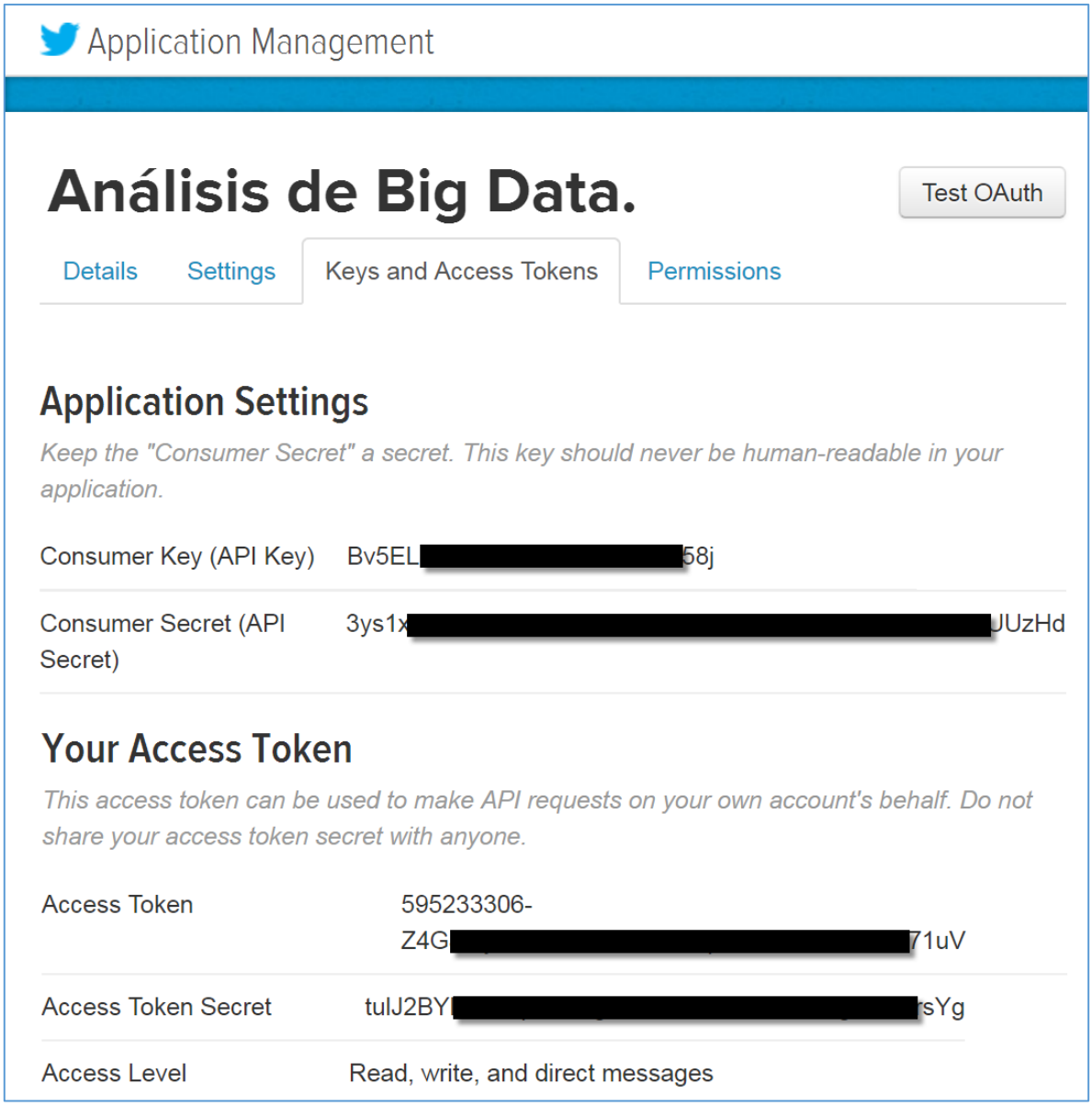


Ilustración 26. Configuración de accesos para la aplicación en twitter.

El servicio que se va a consumir dentro de la *API*, se llama *twitter search*, el cual permite realizar consultas de los tuits recientes o populares, similar a lo que ofrece la interface web y móvil. La *API* consulta una muestra de los tuits publicados de los últimos 7 a 9 días, y permite añadir operadores para filtrar los resultados, como por ejemplo buscar una palabra específica, varias palabras en desorden o una frase exacta, rangos de fechas, geolocalización e idioma.

En la Ilustración 27 se muestra un ejemplo de cómo realizar la consulta de tuits utilizando la *API*, para fines demostrativos, se realiza la consulta de tuits que contengan la palabra UNAM, se sitúe en la ciudad de México por medio de latitud y longitud con un radio de 10 kilómetros, y el idioma de la publicación sea en español.

- En la línea 23 se le pasa como parámetro la palabra *UNAM*.
- En la línea 24 y 25 se define la Latitud en *19.432582* y Longitud de *-99.133090* respectivamente, la cual es la ubicación de la Ciudad de México.
- En la línea 26 se define el radio de 10, y en la línea 32 se define que la unidad del radio será en kilómetros.
- En la línea 34 se define como máximo 1 resultado para este ejemplo.
- En la línea 37 se especifica el idioma español con *es*.
- De la línea 43 a 67 se imprimen los datos que se definieron en el análisis de la sección 3.1.3 como útiles para el análisis de *Big Data*.

```
12 public class QueryExample {
13
14     public static void main(String[] args) throws TwitterException {
15
16         // Realiza la configuración obteniendo las credenciales de acceso
17         // al API contenidas en el archivo twitter4j.properties
18         ConfigurationBuilder cb = new ConfigurationBuilder();
19         TwitterFactory tf = new TwitterFactory(cb.build());
20         Twitter twitter = tf.getInstance();
21
22         // Recibe el string para filtrar la busqueda.
23         Query query = new Query("UNAM");
24         double latitude = 19.432582;
25         double longitude = -99.133090;
26         double radius = 10;
27
28         // Recibe latitud y longitud para buscar por zona geográfica específica.
29         query.setGeoCode(
30                 new GeoLocation(latitude, longitude),
31                 radius,
32                 Query.KILOMETERS);
33         // Delimita el número de tuits de respuesta de 1 a máximo 100.
34         query.count(1);
35
36         // Filtramos para tuits unicamente en Español, Inglés o cualquier idioma.
37         query.setLang("es");
38
39         // Ejecutamos el metodo que realiza la consulta.
40         QueryResult result = twitter.search(query);
41
42         // Iteramos los resultados
43         for (Status status : result.getTweets()) {
44             System.out.println("---Datos Del Tuit:");
45             System.out.println("Texto del tuit: " + status.getText());
46             System.out.println("Fecha de publicación: " + status.getCreatedAt());
47             System.out.println("Id: " + status.getId());
48             System.out.println("Si es re-tuit: " + status.getRetweetCount());
49             System.out.println("---Datos del Usuario:");
50             System.out.println("Nombre del Usuario: " + status.getUser().getName());
51             System.out.println("Localización del Usuario: " + status.getUser().getLocation());
52             System.out.println("Id del Usuario: " + status.getUser().getId());
53             System.out.println("Numero de Seguitdores del Usuario: " + status.getUser().getFollowersCount());
54             System.out.println("Descripción del Usuario: " + status.getUser().getDescription());
55             System.out.println("Numero de Contactos del Usuario: " + status.getUser().getFriendsCount());
56             System.out.println("Nombre Público del Usuario: " + status.getUser().getScreenName());
57             System.out.println("---Hashtags:");
58
59             // Verifica si el tuit contiene más de cero Hastags en su contenido.
60             if (status.getHashtagEntities().length > 0) {
61                 HashtagEntity[] hashtagEntity = status.getHashtagEntities().clone();
62                 // Iteramos los Hashtags
63                 for ( HashtagEntity o: hashtagEntity ){
64                     System.out.println("---Hashtag: "+ o.getText());
65                 }
66             }
67         }
68     }
69 }
```

Ilustración 27. Consulta de tuits con parámetros definidos de búsqueda.

Como resultado de la ejecución del ejemplo, podemos observar en la Ilustración 28 la impresión en la consola de los datos que se utilizarán en el análisis, donde podemos ver el texto del tuit, la fecha de publicación, el identificador único del tuit y el número de veces que ha sido retuiteado. Se ven también los datos como el nombre del usuario, su localización registrada en su perfil, el identificador del usuario único, el número de seguidores, la descripción del usuario, número de contactos del usuario y nombre público del usuario. Al final se imprimen los *hashtags* contenidos en el tuit.

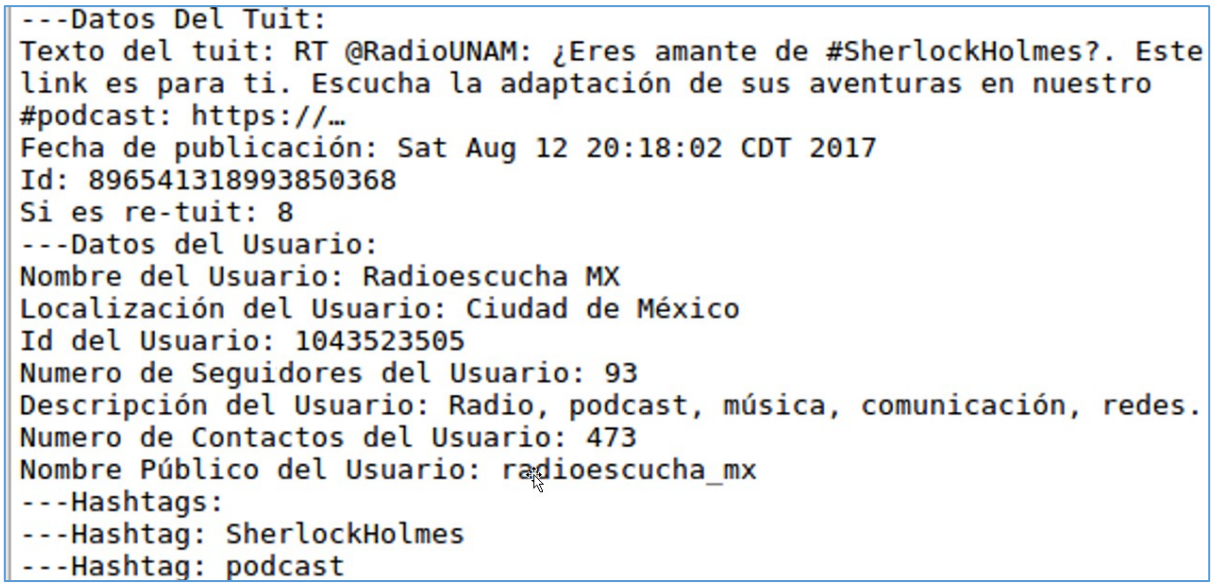


Ilustración 28. Resultado de la consulta de la API.

4.1.2 Análisis de sentimiento con la biblioteca de Stanford NLP

En la sección 1.5 del marco teórico, se habló de la biblioteca de Stanford NLP, para realizar el análisis de sentimiento, por lo que en esta sección se mostrará su implementación.

En la Ilustración 29 se muestra el método *getSentiment()* que devuelve el sentimiento de una cadena de texto en inglés:

1. En la línea 13 vemos que se le pasa como argumento la cadena de caracteres.
2. En la línea 20 se configura el análisis de sentimiento que será realizado.
3. En la línea 28 se pasa la cadena a ser analizada al constructor de la clase Annotation para ser analizada.
4. En la línea 31 se ejecuta el análisis de sentimiento de la cadena.
5. En la línea 39 se almacena el sentimiento obtenido en la variable *sentiment* que devolverá el método.

```
1  package Stanford;
2
3  import java.util.*;
4
5  import edu.stanford.nlp.ling.*;
6  import edu.stanford.nlp.pipeline.*;
7  import edu.stanford.nlp.sentiment.SentimentCoreAnnotations;
8  import edu.stanford.nlp.util.*;
9
10 /** This class demonstrates building and using a Stanford CoreNLP pipeline. */
11 public class StanfordSentimentAnalyzer {
12
13   public static String getSentiment(String line) {
14
15         String sentiment = null;
16
17         // creates a StanfordCoreNLP object, with POS tagging,
18         // lemmatization, NER, parsing, and coreference resolution
19         Properties props = new Properties();
20         props.setProperty("annotators",
21                 "tokenize, ssplit, pos, lemma, ner, parse, dcoref, sentiment");
22
23         StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
24
25         // Initialize an Annotation with some text to be annotated.
26         // The text is the argument to the constructor.
27         Annotation annotation;
28         annotation = new Annotation(line);
29
30         // run all the selected Annotators on this text
31         pipeline.annotate(annotation);
32
33         // An Annotation is a Map with Class keys for the linguistic analysis types.
34         // You can get and use the various analyses individually.
35         // For instance, this gets the parse tree of the first sentence in the text.
36         List<CoreMap> sentences = annotation.get(CoreAnnotations.SentencesAnnotation.class);
37         if (sentences != null && ! sentences.isEmpty()) {
38           CoreMap sentence = sentences.get(0);
39           sentiment = sentence.get(SentimentCoreAnnotations.SentimentClass.class);
40         }
41
42         return sentiment;
43   }
44
45 }
46
```

Ilustración 29. Método que devuelve el sentimiento de una cadena de texto en inglés.

La biblioteca nos puede devolver cinco tipos de sentimiento: Muy Negativo, Negativo, Neutral, Positivo y Muy Positivo, como se explicó en la sección 1.5.

Para realizar un ejemplo, se utilizará la frase analizada en la Ilustración 3 de la sección 1.5, en inglés, puesto que el análisis será realizado en Estados Unidos de Norteamérica.

"This movie doesn't care about cleverness, wit or any other kind of intelligent humor."

En la Ilustración 30, muestra el resultado del análisis de sentimiento que nos devuelve el método, es *negativo*.

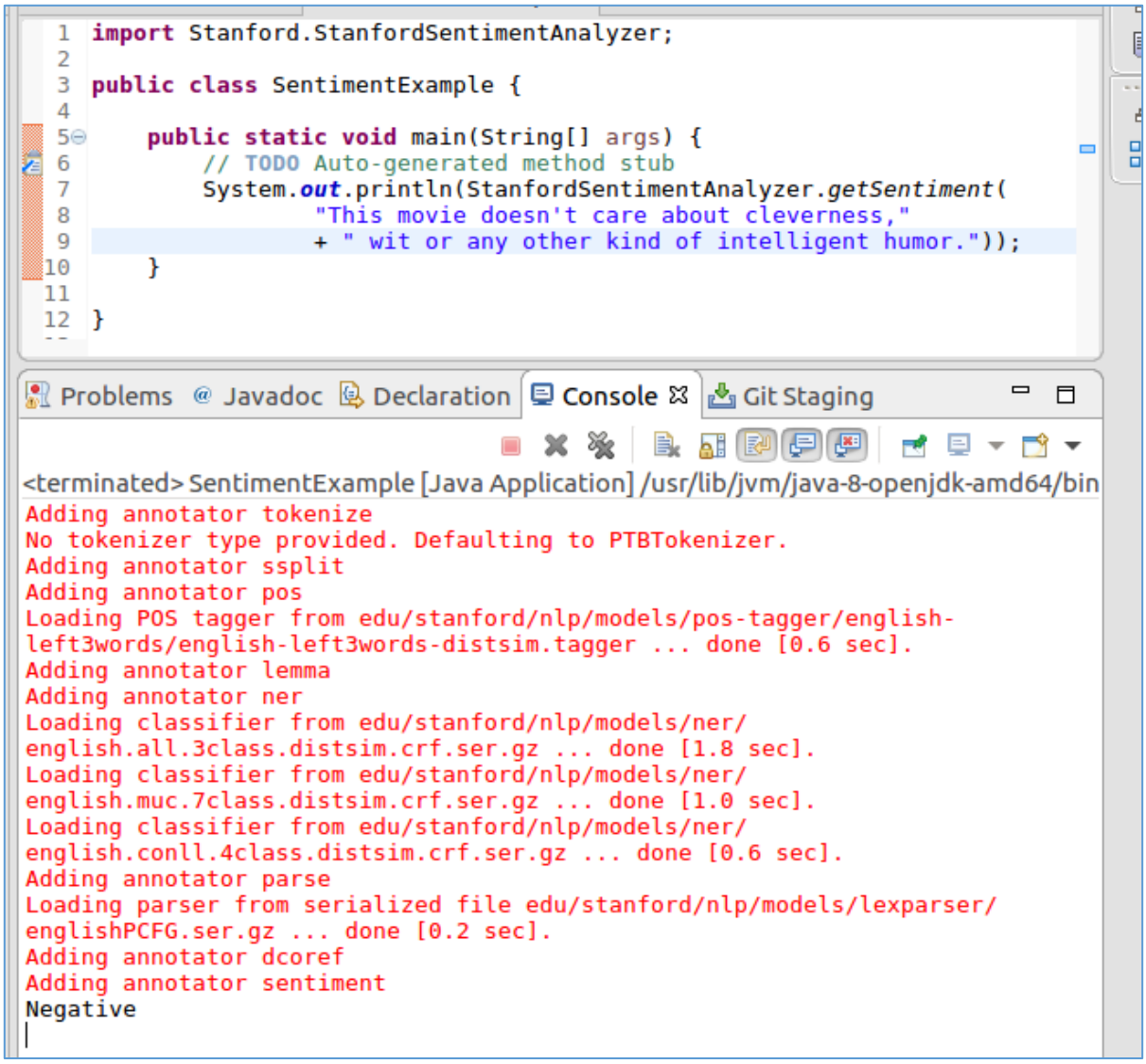


Ilustración 30. Ejecución del método para extraer el sentimiento de una oración.

Realizando un ejemplo de cómo integrar el análisis de sentimiento en un tuit en inglés, se invoca el método como se muestra en la Ilustración 31, en donde observamos lo siguiente:

1. En la línea 37 se configuró para que sea en inglés el tuit, ya que el analizador de sentimiento está hecho para realizar el análisis en el idioma inglés.
2. En la línea 49 podemos ver que se manda a imprimir en la consola el resultado del análisis de sentimiento.

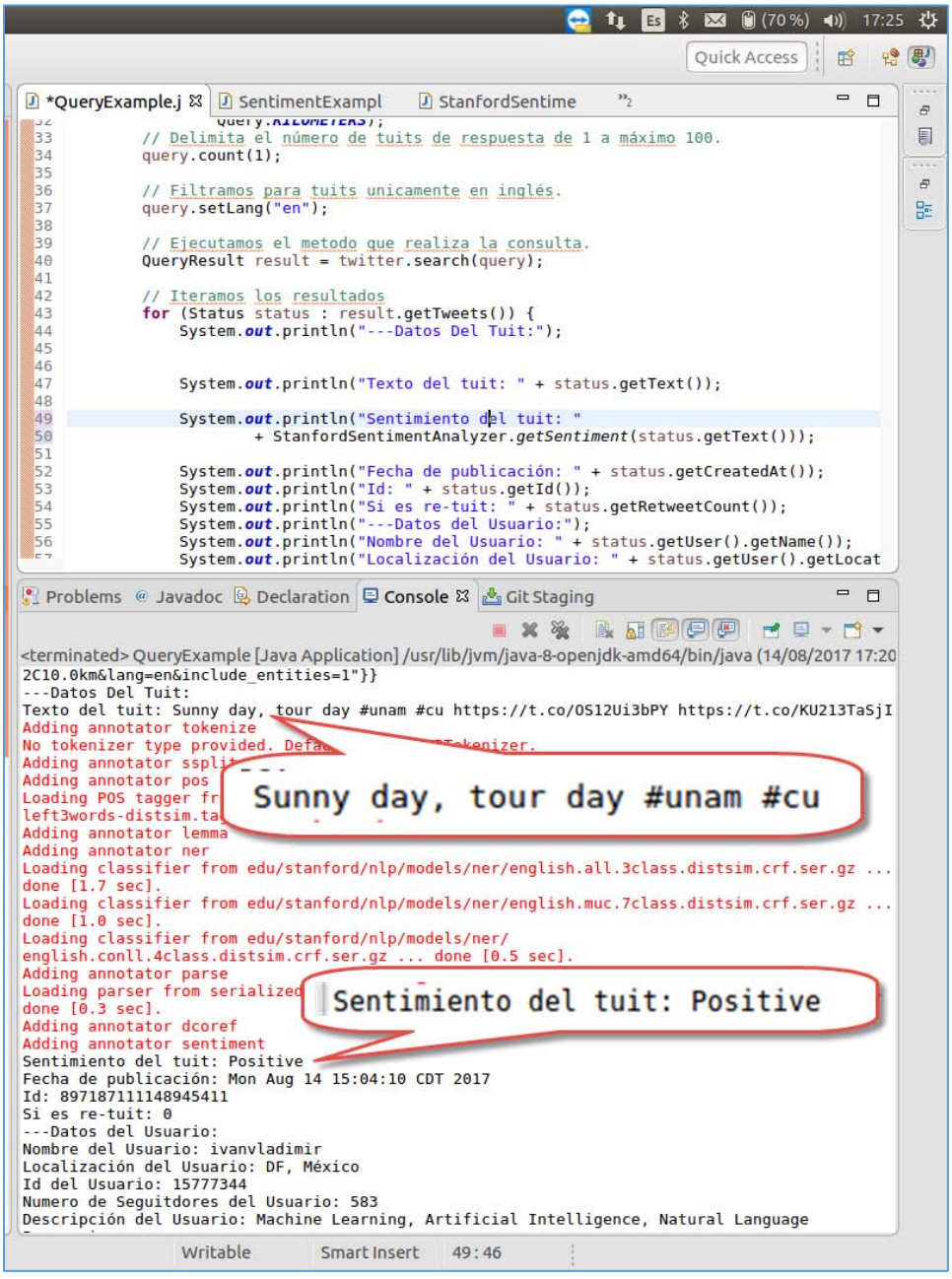


Ilustración 31. Análisis de sentimiento de un tuit.

4.2 Validación de los datos y limpieza

La validación de los datos requiere un procesamiento en tiempo real, por lo que la limpieza se realizará en la memoria principal antes de ser almacenada en la memoria secundaria de la base de datos.

Se definirá el modelo de la base de datos y cómo será la relación entre sus tablas, se describirán los paquetes que integran el programa de procesamiento de tuits, la arquitectura del hardware que se implementará, la codificación del programa y la secuencia de la ejecución del mismo.

4.2.1 Requerimientos del procesamiento de datos

Una vez entendido como consultar un tuit y obtener el sentimiento del mismo, se requiere que el programa realice las siguientes operaciones:

1. Que realice las consultas de tuits sobre las 10 ciudades más pobladas de los Estados Unidos de Norteamérica, que se muestran en la Ilustración 32, filtrando la ubicación por latitud y longitud, y un radio de 15 kilómetros.
2. Que se ejecute cada sesenta segundos posterior a la última búsqueda de tuits realizada en la décima ciudad, consultando y almacenando los últimos 15 tuits de cada ciudad.
3. Que descarte todos los que sean re-tuit, que son aquellos que el usuario decide tomar de una publicación de otro usuario, para compartirla entre sus contactos.
4. Que almacene de los tuits, los datos del perfil de cada usuario y los *hashtags* contenidos en cada tuit.
5. Almacenará únicamente los tuits en inglés.
6. Utilizará el catálogo de la sección 3.1.2 de la Tabla 8 que contiene las palabras relacionadas con drogas, para así identificar qué tuits las contienen.
7. Utilizar una expresión regular para conservar únicamente los caracteres válidos, ya que los datos provenientes de twitter, contienen caracteres no reconocibles por el analizador de sentimiento.

ciudad	estado	latitud	longitud	censo_2010	radio
New York	New York	40.66430	-73.93850	8175133	15
Los Angeles	California	34.01940	-118.41080	3792621	15
Chicago	Illinois	41.83760	-87.68180	2695598	15
Houston	Texas	29.78050	-95.38630	2100263	15
Philadelphia	Pennsylvania	40.00940	-75.13330	1526006	15
Phoenix	Arizona	33.57220	-112.08800	1445632	15
San Antonio	Texas	29.47240	-98.52510	1327407	15
San Diego	California	32.81530	-117.13500	1307402	15
Dallas	Texas	32.77570	-96.79670	1197816	15
San Jose	California	37.29690	-121.81930	945942	15

Ilustración 32. Las diez ciudades más pobladas con su estado, latitud, longitud, el radio de la búsqueda (U.S. Census Bureau, 2010).

4.2.2 Base de datos

Una vez identificados las entidades y los datos disponibles, estos se almacenarán en la base de datos que comprende la descripción de la Tabla 9:

Tabla	Contenido
twitter_user	Contiene los datos generales de los usuarios.
twitter_tweets	Contiene las publicaciones hechas por los usuarios con sus datos generales.
twitter_*hashtags*	Contiene todos los *hashtags* que contienen cada uno de los tuits.
Ciudades	Contiene los datos necesarios de las ciudades en las que se aplica la consulta de tuits.
twitter_filtro	Contiene el identificador del grupo de palabras que se buscará en los tuits.
twitter_filtro_palabras	Contiene todas las palabras que serán buscadas en los tuits.
twitter_tweets_filtro_palabra	Contiene el identificador de las palabras que se encontraron en los tuits.

Tabla 9. Descripción de las tablas.

El modelo de base de datos de la sección 3.1.3 en la Ilustración 21 que se diseñó, será adaptado a las herramientas para procesar *Big Data*, ya que éstas requieren de un identificador numérico auto incrementable, consecutivo y único como llave primaria en cada una de las tablas, para realizar la sincronización de datos entre el esquema relacional y el de archivos de Hadoop, que se verá más adelante, quedando como se muestra en la Ilustración 33.

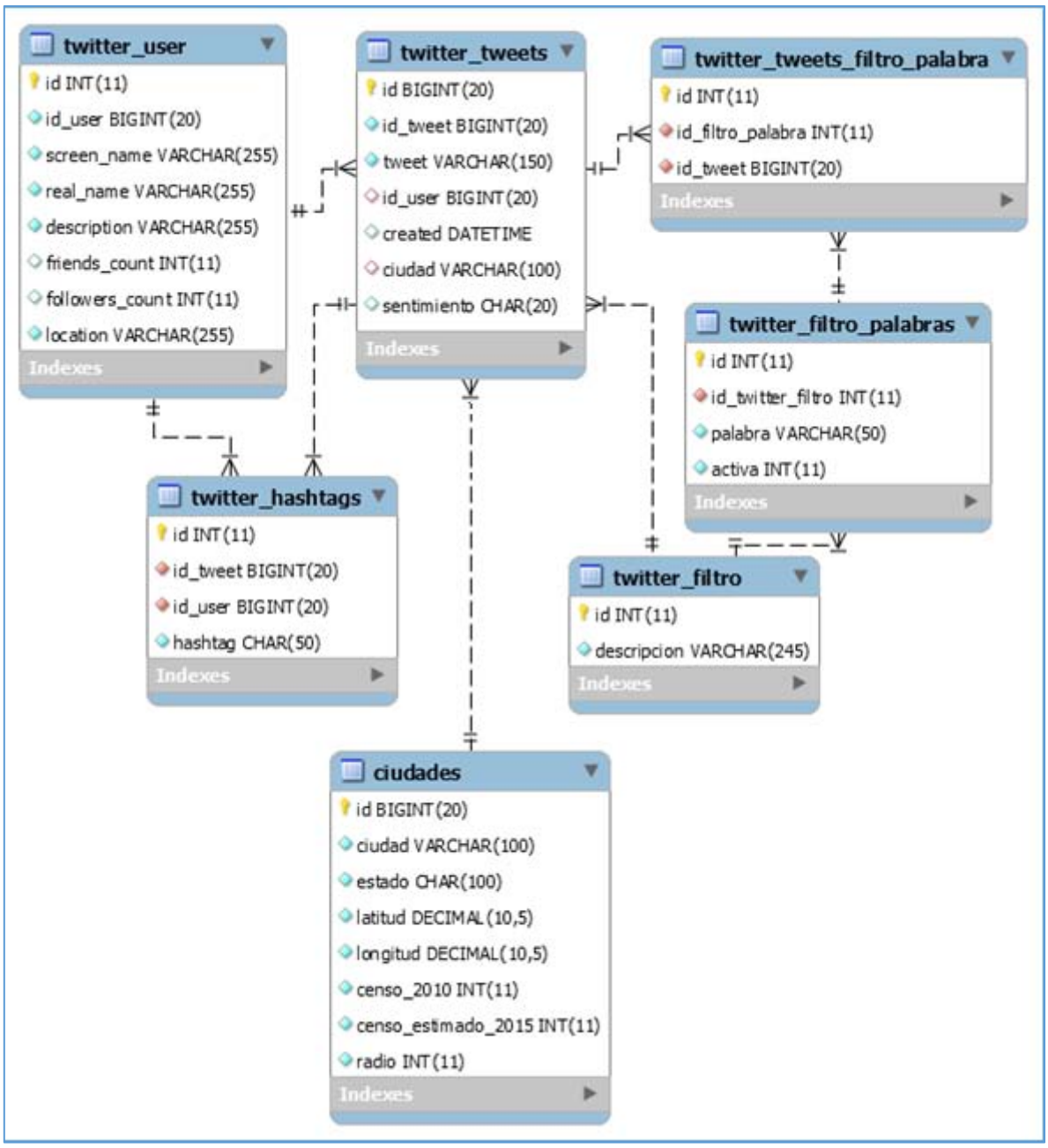


Ilustración 33. Diagrama relacional para el almacenamiento de datos.

Utilizaremos como manejador de base de datos a My*SQL*, elegido dadas sus características como su gran presencia en la industria, es software libre y tiene una gran facilidad de uso.

4.2.3 Paquetes de la arquitectura del programa

Basándose en el patrón *Modelo-Vista-Controlador (MVC),* utilizado frecuentemente en aplicaciones web, para la aplicación, se utilizarán los conceptos de *Controlador* y *Modelo*, y se añadirá un elemento conocido como *Servicio*, que son métodos especializados que se agregan utilizando librerías fuera del marco de trabajo definido. No se considerará la *Vista*, ya que esta aplicación no tiene ninguna interacción con ningún usuario final.

Tenemos tres paquetes definidos para el programa que obtendrá el flujo de datos en tiempo real, como se muestra en la Ilustración 34, y se describen más adelante:

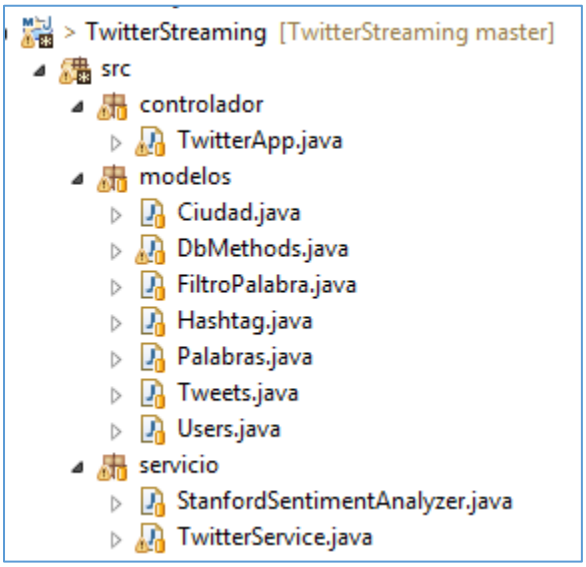


Ilustración 34. Paquetes y clases del proyecto para el streaming de datos de twitter.

1. *Modelos*: En este paquete que se puede observar en la Ilustración 35, se tiene mapeada la base de datos en clases por cada una de sus tablas, esto para poder realizar la manipulación de tipos de datos al momento de consultar, insertar, borrar o actualizar en la base de datos.

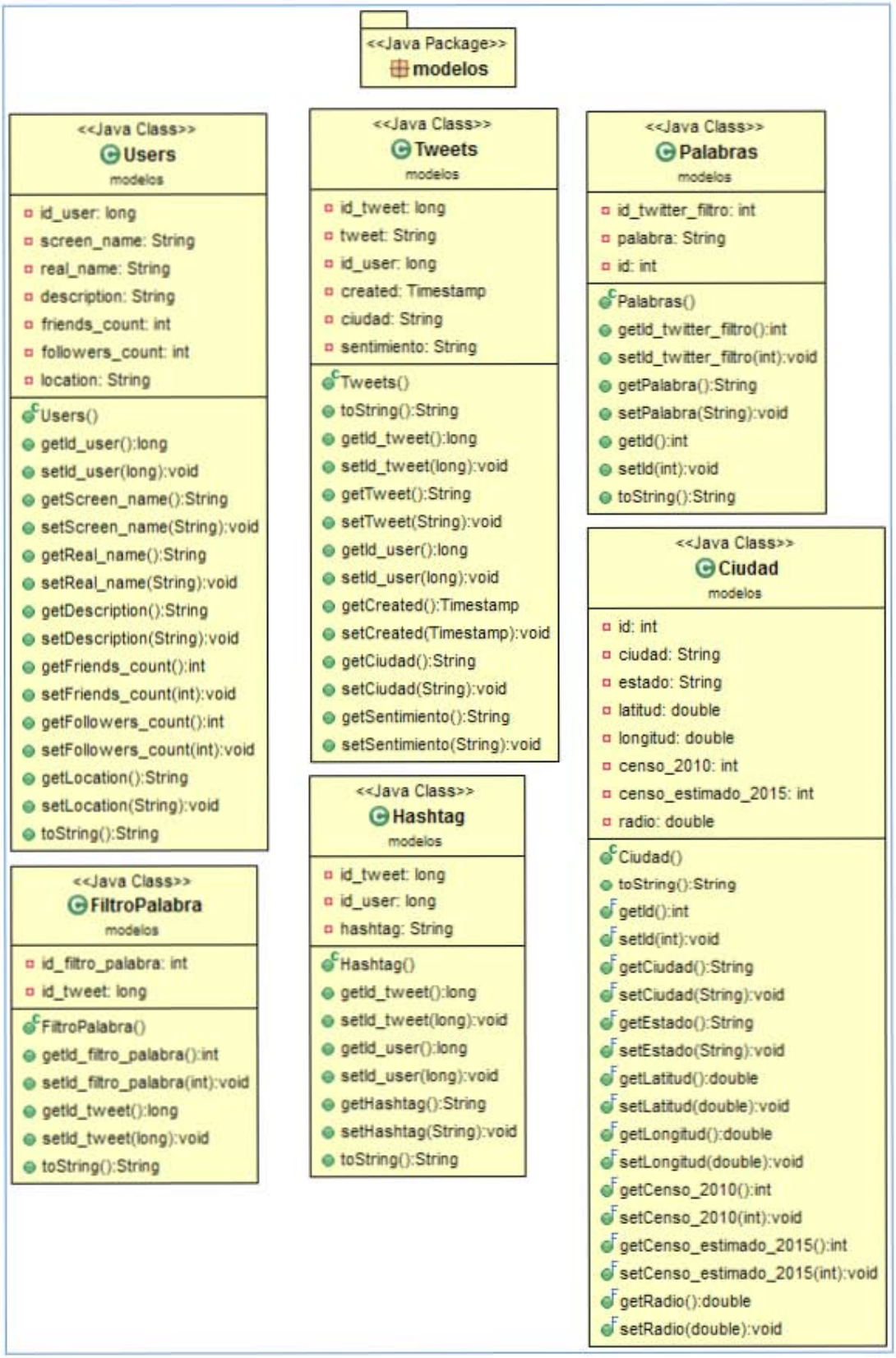


Ilustración 35. Diagrama de clases del paquete de modelos.

Se tiene adicionalmente una clase con los métodos para la manipulación de los datos como lo muestra la Ilustración 36, necesarios para la interacción con la base de datos y las clases contenidas en el *modelo*.

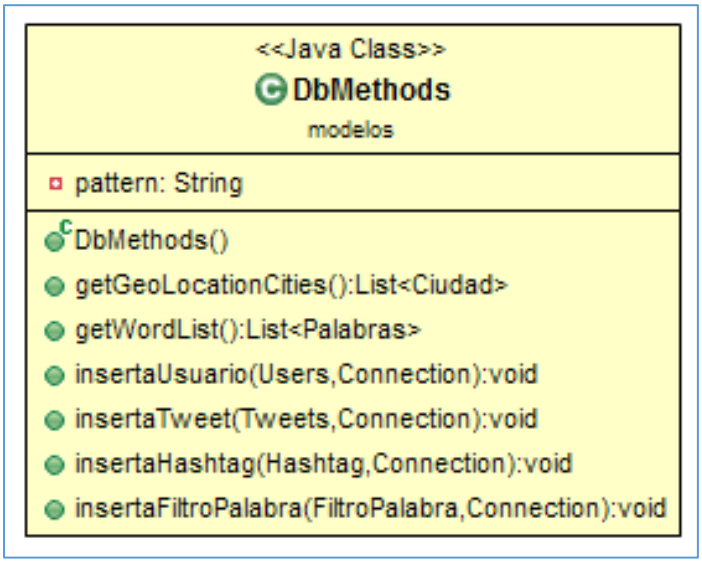


Ilustración 36. Métodos que interactúan con la base de datos.

2. *Controlador*: En este paquete se contiene una única acción como lo muestra la Ilustración 37 en el diagrama de clases, la cual básicamente inicia la ejecución de las consultas de tuits mediante los servicios integrados en el programa.

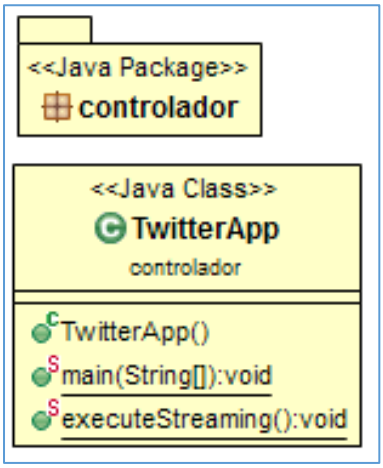


Ilustración 37. Paquete del controlador con su clase que contiene la acción.

3. *Servicios*: En este paquete contenemos los métodos utilizados como se muestra en la Ilustración 38, los cuales consumen las bibliotecas necesarias para poder obtener el flujo de datos directamente de twitter, y obtener el sentimiento de cada tuit publicado de cada usuario en tiempo real.

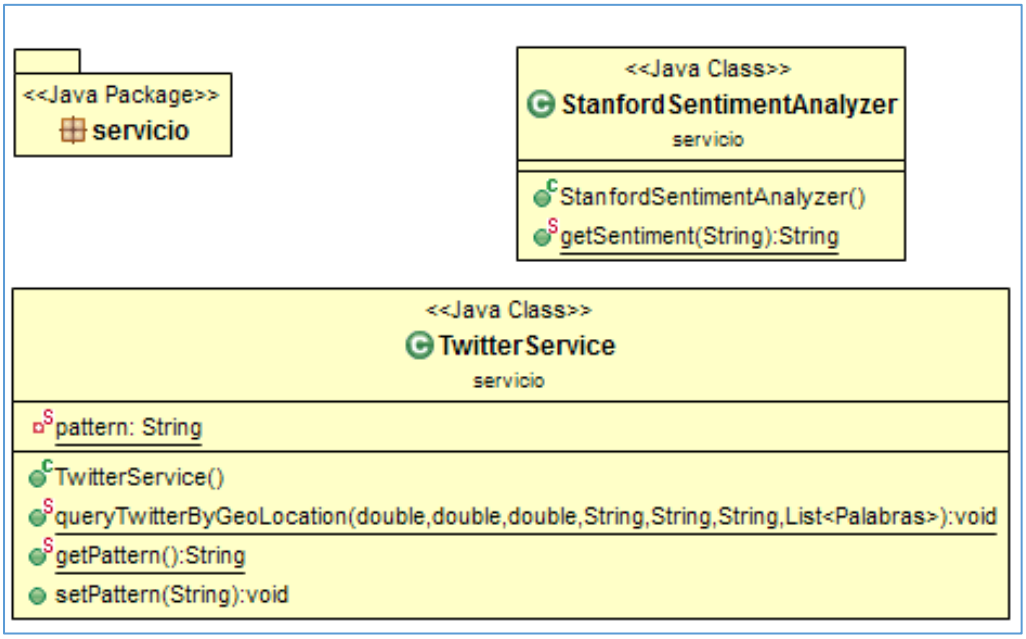


Ilustración 38. Paquete de servicio con sus respectivas clases.

4.2.4 Infraestructura a utilizar

Se utilizaron servicios en la nube que provee Amazon AWS, con el objetivo de poder tener un servicio en ejecución las 24 horas del día, los 7 días de la semana, para tener un flujo de datos en tiempo real.

La Infraestructura en donde se ejecutará la aplicación cuenta con las siguientes características:

Servidor de Aplicación.

- Instancia t2.small de Amazon.
- Memoria de 2 GB.
- Sistema Operativo Ubuntu 16.04LTS.
- Disco Duro de 30 GB.
- My*SQL* 5.7.

4.2.5 Codificación y ejecución del programa

El programa codificado en la Ilustración 39 sigue la siguiente secuencia:

1. En la línea 29 se obtiene la lista de las diez ciudades más pobladas.
2. En la línea 31 se obtienen la lista de las palabras que se buscarán en los tuits.
3. En la línea 44 se iteran las ciudades.
4. En la línea 49 se ejecuta el método *queryTwitterByGeoLocation* que se muestra en la Ilustración 40 que se describirá más adelante, con los argumentos de la ciudad, el cual

realiza la consulta mediante el *API* de twitter, procesa los tuits y los almacena en la base de datos de My*SQL*.

```
 * @author Juan Garfias Vázquez.
 * @version 1.0
 *
 */
public class TwitterApp {

    @SuppressWarnings("static-access")
    public static void main(String args[]) throws Exception {
        executeStreaming();
    }

    @SuppressWarnings("static-access")
    public static void executeStreaming () throws Exception{
        TwitterMethods twitterMethods = new TwitterMethods();
        DbMethods dbMethods = new DbMethods();
        // Obtiene la lista de ciudades.
        List<Ciudad> result = dbMethods.getGeoLocationCities();
        // Obtiene la lista de palabras que serán buscadas en el tuit.
        List<Palabras> palabras = dbMethods.getWordList();
        // Obtengo los datos de configuración para el programa.
        Properties props = new Properties();
        FileInputStream fis = null;
        fis = new FileInputStream("db.properties");
        props.load(fis);

        Class.forName(props.getProperty("driver"));

        // Ciclo infinito de ejecución del programa.
        do{
            Date date = new Date();
            // Iteración de las 10 ciudades consultadas.
            for ( Ciudad ciudad : result ) {
                //sleep 5 seconds
                try {
                    // Ejecutamos el método para la consulta de los tuits.
                    // Recibe los argumentos para configurar la consulta de tuits.
                    twitterMethods.queryTwitterByGeoLocation(
                            ciudad.getLatitud(),
                            ciudad.getLongitud(),
                            ciudad.getRadio(),
                            "",
                            ciudad.getCiudad(),
                            "en",
                            palabras);  // Recibe la lista de palabras a buscar.
                    System.out.println("Se realizó la consulta de la Ciudad: "
                            + ciudad.getCiudad());
                } catch (Exception e) {
                    // TODO: handle exception
                    System.out.println("Excedido el limite de consultas, esperando...");
                    System.out.println(e);
                    Thread.sleep(300000);
                }
            }

            // Indica el tiempo de espera de 60 segundas para la siguiente iteración.
            Thread.sleep(60000);
        } while (true);
    }
}
```

Ilustración 39. Codificación del programa en la etapa de iteración de las ciudades para consulta de tuits.

En la Ilustración 40 vemos una sección del método *queryTwitterByGeoLocation()*, en donde se realiza la consulta y validación principal de los tuits:

5. En la línea 84, se iteran los tuits de resultado por la consulta con el *API*, los cuales como recordaremos, son 15 resultados por ciudad. Por lo tanto, se procesan por cada ejecución del programa, 150 tuits cada 60 segundos.
6. En la línea 85 se evalúa que un tuit no sea retuit.
7. En la línea 94 obtenemos el sentimiento del tuit.
8. En la líneas 107,108 y 135 se inserta el usuario de twitter, el tuit y los *hashtags* que contenga el tuit.
9. En las líneas 109 a 125, se realiza la búsqueda de las palabras en los tuits, y al ser encontrada una palabra, esta se almacena en la base de datos.

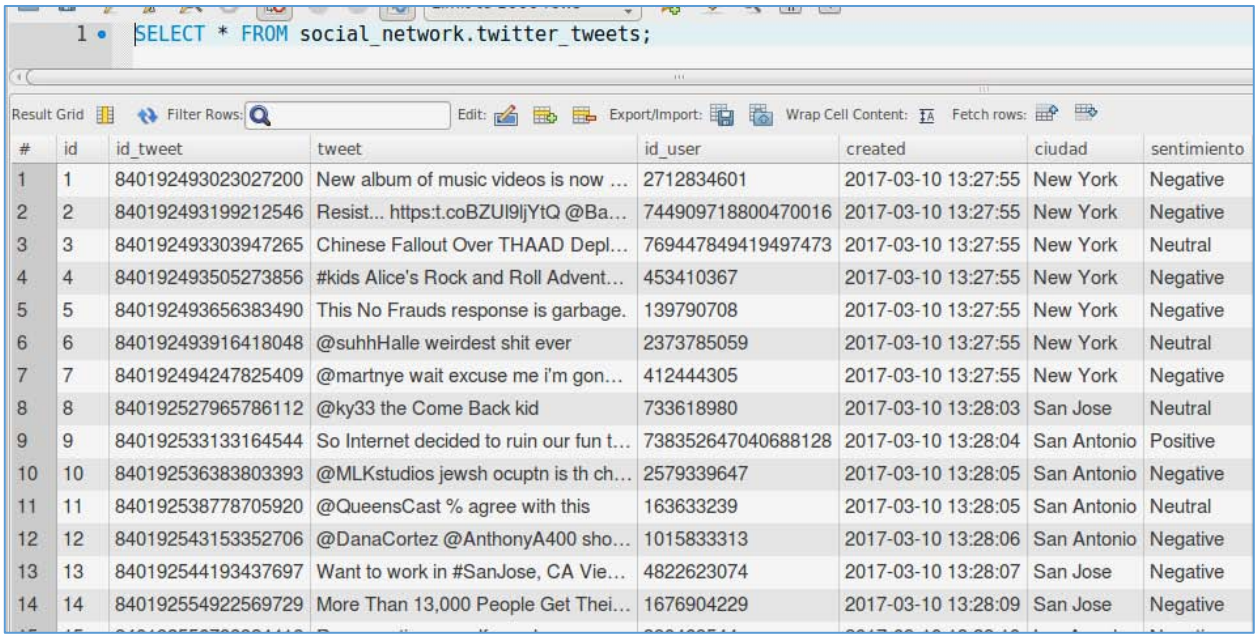
```
83
84          for (Status status : result.getTweets()) {
85              if (status.getRetweetCount() == 0) {
86                  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
87                  @SuppressWarnings("unused")
88                  String date = sdf.format(status.getCreatedAt());
89                  java.sql.Timestamp sqlDate = new java.sql.Timestamp( status.getCreatedAt().getTime() );
90                  tweet.setCiudad(ciudad);
91                  tweet.setId_tweet(status.getId());
92                  tweet.setId_user(status.getUser().getId());
93                  tweet.setTweet(status.getText());
94                  tweet.setSentimiento(
95                          StanfordSentimentAnalyzer.getSentiment(
96                                  status.getText().replaceAll( getPattern()  , "").trim() )
97                          );
98                  tweet.setCreated( sqlDate );
99                  // Datos del Usuario.
100                 user.setId_user(status.getUser().getId());
101                 user.setScreen_name(status.getUser().getScreenName());
102                 user.setReal_name(status.getUser().getName());
103                 user.setDescription(status.getUser().getDescription());
104                 user.setFriends_count(status.getUser().getFriendsCount());
105                 user.setFollowers_count(status.getUser().getFollowersCount());
106                 user.setLocation(status.getUser().getLocation());
107                 dbMethods.insertaUsuario(user, c);
108                 dbMethods.insertaTweet(tweet, c);
109                 String s = status.getText().toLowerCase();
110                 for (Palabras palabra : palabras){
111                     int validaString = s.indexOf(" "+palabra.getPalabra()+" ");
112                     if(validaString>0){
113                         FiltroPalabra filtroPalabra = new FiltroPalabra();
114                         filtroPalabra.setId_filtro_palabra(palabra.getId_twitter_filtro());
115                         filtroPalabra.setId_tweet(status.getId());
116                         dbMethods.insertaFiltroPalabra(filtroPalabra, c);
117                     }
118                     validaString = s.indexOf("#"+palabra.getPalabra()+" ");
119                     if(validaString>0){
120                         FiltroPalabra filtroPalabra = new FiltroPalabra();
121                         filtroPalabra.setId_filtro_palabra(palabra.getId_twitter_filtro());
122                         filtroPalabra.setId_tweet(status.getId());
123                         dbMethods.insertaFiltroPalabra(filtroPalabra, c);
124                     }
125                 }
126                 // Datos del Hashtag
127                 if (status.getHashtagEntities().length > 0) {
128                     HashtagEntity[] hashtagEntity = status.getHashtagEntities().clone();
129                     for ( HashtagEntity o: hashtagEntity ){
130                         //System.out.println(o.getText());
131                         hashtag.setId_tweet(status.getId());
132                         hashtag.setId_user(status.getUser().getId());
133                         hashtag.setHashtag(o.getText());
134
135                         dbMethods.insertaHashtag(hashtag, c);
136                     }
137                 }
138             }
139         }
140         c.commit();
141         st.close();
```

Ilustración 40. Iteración de tuits y almacenamiento en la base de datos.

El código fuente de esta aplicación se encuentra publicado en Github en la siguiente ruta: *https://github.com/Juanin88/TwitterStreaming*

Una vez ejecutado el programa, observamos los datos almacenados en las tablas de la base de datos de My*SQL*.

En la Ilustración 41, se muestra la tabla que contiene los tuits, en donde se puede observar que contiene un *id*, el cual se agregó como un dato auto incrementable para poder realizar la sincronización de los datos que se verá más adelante, se tiene el *id_tweet*, el cual es el identificador único de cada tuit, *tweet* que es el contenido de la publicación del tuit, el *id_user*, que es el usuario que realizó el tuit, *created* que es la fecha y hora en que se publicó el tuit, *ciudad* la cual se asignó al momento de realizar la iteración de consulta de tuits por ciudades y por último, el *sentimiento* obtenido del tuit.



```
1 • SELECT * FROM social_network.twitter_tweets;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

#	id	id_tweet	tweet	id_user	created	ciudad	sentimiento
1	1	840192493023027200	New album of music videos is now …	2712834601	2017-03-10 13:27:55	New York	Negative
2	2	840192493199212546	Resist... https:t.coBZUl9ljYtQ @Ba…	744909718800470016	2017-03-10 13:27:55	New York	Negative
3	3	840192493303947265	Chinese Fallout Over THAAD Depl…	769447849419497473	2017-03-10 13:27:55	New York	Neutral
4	4	840192493505273856	#kids Alice's Rock and Roll Advent…	453410367	2017-03-10 13:27:55	New York	Negative
5	5	840192493656383490	This No Frauds response is garbage.	139790708	2017-03-10 13:27:55	New York	Negative
6	6	840192493916418048	@suhhHalle weirdest shit ever	2373785059	2017-03-10 13:27:55	New York	Neutral
7	7	840192494247825409	@martnye wait excuse me i'm gon…	412444305	2017-03-10 13:27:55	New York	Negative
8	8	840192527965786112	@ky33 the Come Back kid	733618980	2017-03-10 13:28:03	San Jose	Neutral
9	9	840192533133164544	So Internet decided to ruin our fun t…	738352647040688128	2017-03-10 13:28:04	San Antonio	Positive
10	10	840192536383803393	@MLKstudios jewsh ocuptn is th ch…	2579339647	2017-03-10 13:28:05	San Antonio	Negative
11	11	840192538778705920	@QueensCast % agree with this	163633239	2017-03-10 13:28:05	San Antonio	Neutral
12	12	840192543153352706	@DanaCortez @AnthonyA400 sho…	1015833313	2017-03-10 13:28:06	San Antonio	Negative
13	13	840192544193437697	Want to work in #SanJose, CA Vie…	4822623074	2017-03-10 13:28:07	San Jose	Negative
14	14	840192554922569729	More Than 13,000 People Get Thei…	1676904229	2017-03-10 13:28:09	San Jose	Negative

Ilustración 41. Contenido de la tabla de tuits

En la Ilustración 42, observamos el *id* auto incrementable necesario para la sincronización de datos, el *id_user* que es el identificador único del usuario de twitter, *screen_name*, que es su nombre de usuario en twitter, *real_name* que es su nombre de usuario real, *description* que es la descripción del usuario, *friends_count* que son los contactos del usuario, *followers_count* que son los seguidores que tiene el usuario y *location* que es la ubicación que registra el usuario.

```
1 • SELECT * FROM social_network.twitter_user;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

#	id	id_user	screen_name	real_name	description	friends_count	followers_count	location
1	1	295	joshk	Josh Kopelman	Socalled venture capitalist. Father. …	3172	114830	Philly
2	2	357	wubbahed	Will Turnage	Lead Technology at Inamoto & Co.…	387	2218	Brooklyn, NY
3	3	418	dens	Dennis Crowley	I like to build things Founder @Fou…	2092	85506	NYC / Kingston
4	4	528	buzz	Buzz Andersen	Tech veteran Apple, Square, Tumb…	1256	15017	NYC
5	5	556	ch	charlie wright	Never for money, always for love. …	779	2769	downtown los angeles, ca
6	6	573	heyitsnoah	Noah Brier	Cofounder of @Percolate	1018	15753	NYC
7	7	744	shahid	yung halal cart	many things but not what you assume	924	1280	nyc
8	8	765	seanbonner	Sean Bonner	Global Director @Safecast. Fellow …	471	12781	Los Angeles, CA
9	9	1027	harry	harry	my husband is a cop	665	7625	New York, USA
10	10	1033	dyfl	Chris Conroy	Editor @DCComics DETECTIVE …	433	4224	Los Angeles, CA
11	11	1084	toomuchnick	Nick Douglas	Soy Boy, Wokester, I have only be…	3591	13189	Brooklyn, NY
12	12	1180	jeeves	Rajiv Sinclair	https:t.coDUevwGL5W7	2091	1057	Chicago
13	13	1513	brianwmniles	Brian Wm. Niles	Founder & Chief Evangelist, @Tar…	116	1259	Havertown, PA
14	14	1661	paulgb	Paul Butler	Lately reobsessed with creative dat…	739	980	NYC
15	15	1742	trevorturk	Trevor Turk	Freelance Programmer, currently w…	176	1465	Chicago, IL

Ilustración 42. Datos de los usuarios de los tuits.

En la Ilustración 43 se muestra la tabla que contiene los *hashtags* , con su *id* auto incrementable para realizar la sincronización de datos, el *id_tweet* que relaciona el hashtag con el tuit y el *id_user* que relaciona el hashtag con el usuario, y el *hashtag* mismo.

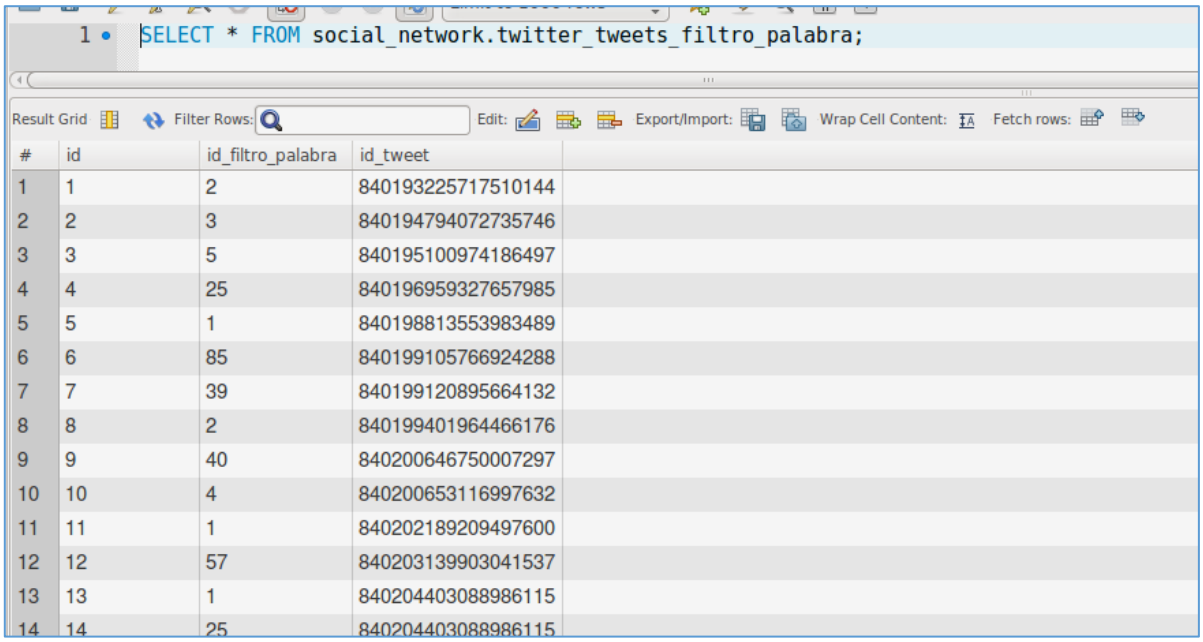
```
1 • SELECT * FROM social_network.twitter_hashtags;
```

Grid | Filter Rows: | Edit: | Export/Imp

id	id_tweet	id_user	hashtag
1	840192493505273856	453410367	kids
2	840192533133164544	738352647040688128	Push
3	840192543153352706	1015833313	SBready
4	840192544193437697	4822623074	Hiring
5	840192544193437697	4822623074	Job
6	840192544193437697	4822623074	Jobs
7	840192544193437697	4822623074	SanJose
8	840192544193437697	4822623074	Transportation
9	840192554922569729	1676904229	education
10	840192561188810756	248135355	np
11	840192561188810756	248135355	SoundCloud
12	840192563403452416	168289712	Facts

Ilustración 43. Hashtags de los tuits.

En la Ilustración 44, se muestra la tabla que contiene la relación entre las palabras encontradas y el tuit que lo contiene, teniendo su *id* auto incrementable para realizar la sincronización, el *id_filtro_palabra* que relaciona la palabra encontrada con el *id_tweet* del tuit.



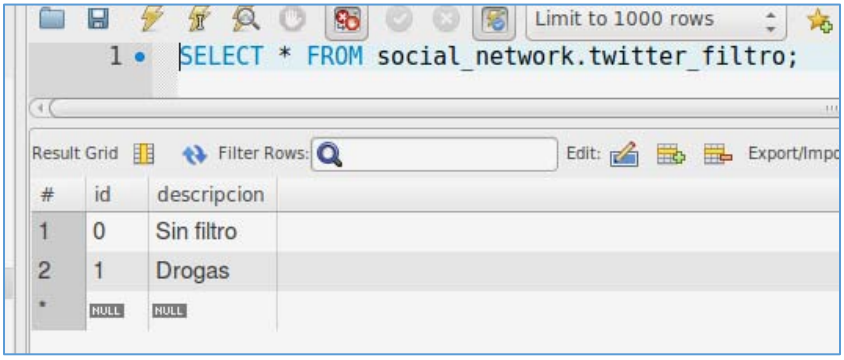
```
1 • SELECT * FROM social_network.twitter_tweets_filtro_palabra;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

#	id	id_filtro_palabra	id_tweet
1	1	2	840193225717510144
2	2	3	840194794072735746
3	3	5	840195100974186497
4	4	25	840196959327657985
5	5	1	840198813553983489
6	6	85	840199105766924288
7	7	39	840199120895664132
8	8	2	840199401964466176
9	9	40	840200646750007297
10	10	4	840200653116997632
11	11	1	840202189209497600
12	12	57	840203139903041537
13	13	1	840204403088986115
14	14	25	840204403088986115

Ilustración 44. Palabra del catálogo encontrada en el tuit.

En la Ilustración 45 se muestra el catálogo que define el tema principal del grupo de palabras, en este caso llamado Drogas, permitiendo de esta manera poder crear más grupos para realizar análisis de cualquier otro dominio.



```
Limit to 1000 rows
1 • SELECT * FROM social_network.twitter_filtro;
```

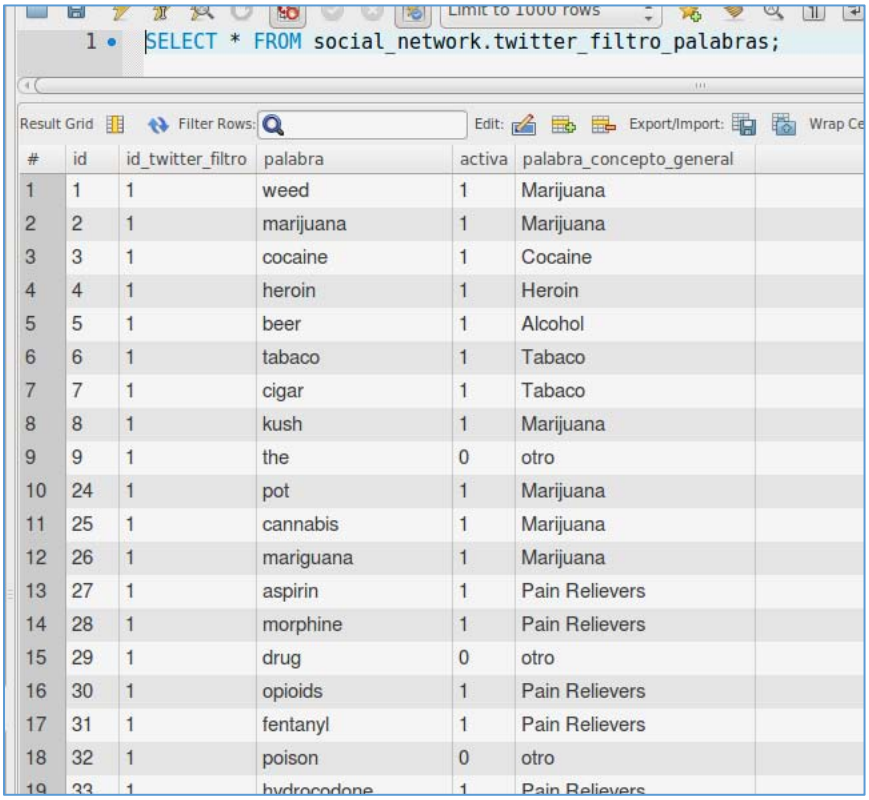
Result Grid | Filter Rows: | Edit: | Export/Impc

#	id	descripcion
1	0	Sin filtro
2	1	Drogas
*	NULL	NULL

Ilustración 45. Tema principal del filtro.

En la Ilustración 46 se muestra el catálogo de palabras relacionadas a un tema, en donde se tiene su *id* auto incrementable, el *id_twitter_filtro* del grupo de palabras, la *palabra*, si la palabra esta *activa.*

Para facilitar el análisis, al existir un sinfín de sinónimos, se agregó un campo llamado *palabra_concepto_general,* que como su nombre lo dice, es el concepto general de los sinónimos que puede haber, ya que como se puede observar en la Ilustración 46, *weed* y *marijuana* se refieren a la misma droga.



```
1 • SELECT * FROM social_network.twitter_filtro_palabras;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

#	id	id_twitter_filtro	palabra	activa	palabra_concepto_general
1	1	1	weed	1	Marijuana
2	2	1	marijuana	1	Marijuana
3	3	1	cocaine	1	Cocaine
4	4	1	heroin	1	Heroin
5	5	1	beer	1	Alcohol
6	6	1	tabaco	1	Tabaco
7	7	1	cigar	1	Tabaco
8	8	1	kush	1	Marijuana
9	9	1	the	0	otro
10	24	1	pot	1	Marijuana
11	25	1	cannabis	1	Marijuana
12	26	1	mariguana	1	Marijuana
13	27	1	aspirin	1	Pain Relievers
14	28	1	morphine	1	Pain Relievers
15	29	1	drug	0	otro
16	30	1	opioids	1	Pain Relievers
17	31	1	fentanyl	1	Pain Relievers
18	32	1	poison	0	otro
19	33	1	hydrocodone	1	Pain Relievers

Ilustración 46. Catálogo de palabras a buscar en el tuit.

4.3 Extracción de los datos

Hasta este momento ya se tienen los datos que se requieren para realizar el análisis, la etapa siguiente consiste en la extracción de los datos contenidos en la base de datos relacional, al ambiente de análisis de *Big Data*, la cual para este caso, la plataforma a utilizar será Hortonworks y el Sistema de Archivos de *Hadoop* (HDF), que se explicó en la sección 0.

4.3.1 Importando datos con *apache* sqoop

Apache Sqoop(TM) es una herramienta lanzada en marzo del 2012, la cual está diseñada para hacer más eficiente la transferencia entre bases de datos relacionales al sistema de archivos de *Apache* Hadoop.

Esta herramienta se utiliza ejecutando comandos desde la terminal, indicando la conexión al origen de los datos. Se utilizarán dos funciones de Sqoop, la primera es la de importar una tabla completa de la base de datos relacional, y la segunda es sincronizar los registros nuevos de la tabla de la base de datos relacional, con la contenida en Hadoop.

A. Importar Tablas

Para la primera función, se tiene la siguiente estructura del comando:

```
$ sqoop import --connect [conexión a la base de datos origen] --username
[Usuario de la base de datos] -P --split-by [indica la columna para
realizar la segmentación] --columns [indica las columnas de la tabla a
importar] --table [nombre de la tabla] --Hive-import --create-Hive-table
--Hive-table [indica el nombre de la base de datos y el nombre de la
tabla que se creará] --driver [indica el conector de la base de datos a
utilizar] -m [número de segmentos (cuatro por default) al descargar]
```

Al importar mediante el script anterior, Sqoop realiza cuatro conexiones paralelas y simultaneas a la base de datos por default, y basándose en el id único que se indica con el argumento –split-by, segmenta en cuatro partes los registros de la tabla, esto hace más eficiente el proceso de importación de los datos.

Por ejemplo, para el caso de nuestra tabla *twitter_tweets* posterior a realizar el proceso de consulta de tuits, contiene 3,182,630 tuits almacenados, por lo que al realizar los cuatro segmentos por default, la memoria de la aplicación se desborda, ya que cada una de las consultas descarga 795,657 tuits, entonces para esa tabla en particular, definimos que se hagan 10 cortes en el argumento *-m* a la tabla para poder realizar la importación de los tuits.

Para importar a *Hadoop* las tablas que se tiene en la base de datos de My*SQL*, los comandos quedarían de la siguiente manera:

Tabla	Script
ciudades	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username **jgarfias** -P --split-by **id** --columns **id,ciudad,estado,latitud,longitud,censo_2010,censo_estimado_2015,radio** --table **ciudades** --*Hive*-import --create-*Hive*-

	table --*Hive*-table **social_network.ciudades** --driver **com.my*SQL*.jdbc.Driver**
twitter_filtro	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username **jgarfias** -P --split-by **id** --columns **id,descripcion** --table **twitter_filtro** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_filtro** --driver **com.my*SQL*.jdbc.Driver**
twitter_filtro_palabras	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username **jgarfias** -P --split-by **id** --columns **id,id_twitter_filtro,palabra,activa,palabra_concepto_general** --table **twitter_filtro_palabras** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_filtro_palabras** --driver **com.my*SQL*.jdbc.Driver**
twitter_*hashtags*	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username jgarfias -P --split-by **id** --columns **id,id_tweet,id_user,hashtag** --table **twitter_*hashtags*** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_*hashtags*** --driver **com.my*SQL*.jdbc.Driver**
twitter_user	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username **jgarfias** -P --split-by **id** --columns **id,id_user,screen_name,real_name,description,friends_count,followers_count,location** --table **twitter_user** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_user** --driver **com.my*SQL*.jdbc.Driver**
twitter_tweets_filtro_palabra	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username jgarfias -P --split-by **id** --columns **id,id_filtro_palabra,id_tweet** --table **twitter_tweets_filtro_palabra** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_tweets_filtro_palabra** --driver **com.my*SQL*.jdbc.Driver**
twitter_tweets	$ sqoop import --connect **jdbc:my*SQL*://127.0.0.1/social_network** --username **jgarfias** -P --split-by **id** --columns **id,id_tweet,tweet,id_user,created,ciudad,sentimiento** --table **twitter_tweets** --*Hive*-import --create-*Hive*-table --*Hive*-table **social_network.twitter_tweets** -m **10** --driver **com.my*SQL*.jdbc.Driver**

B. Sincronizar tablas

Para el proceso de sincronizar las tablas, se genera un comando similar al explicado anteriormente, salvo que este no creará la tabla sino que le agregará los registros nuevos, quedando de la siguiente manera:

```
$ sqoop import --connect [conexión a la base de datos origen] --username [Usuario de la base de datos] --password [password de la base de datos] --split-by [indica la columna para realizar los segmentos]--columns [indica las columnas de la tabla a importar] --table [nombre de la tabla] --Hive-import --Hive-table [indica el nombre de la base de datos y el nombre de la tabla importada] --incremental [indica el timpo de importación] --check-column [indica la columna a comparar el último id] --last-value [máximo id en nuestra tabla de hadoop] --driver [indico el conector de la base de datos a utilizar]
```

Basándose en el ejemplo anterior, ahora el comando para sincronizar la tabla *twitter_tweets* queda de la siguiente manera:

```
$ sqoop import --connect jdbc:mySQL://192.168.0.3/social_network --username root --password root --split-by id --columns id,id_tweet,tweet,id_user,created,ciudad,sentimiento,id_twitter_filtro,filtro_valido --table twitter_tweets --Hive-import --Hive-table social_network.twitter_tweets --incremental append --check-column id --last-value 1902024 --driver com.mySQL.jdbc.Driver
```

Al efectuar el comando anterior, Sqoop realiza una conexión a la base de datos de My*SQL*, consultando el máximo identificador de la tabla, y se le pasa como argumento el máximo identificador que tenemos almacenado la tabla de Hadoop. Posteriormente, obtiene la diferencia sobre los máximos identificadores, para después consultar y descargar la diferencia a la tabla de Hadoop.

Poniendo un ejemplo, en la tabla de *Hadoop* se tienen 10 tuits y su máximo identificador es 10, y en la tabla de My*SQL* se tienen 30 registros, por lo que su máximo identificador en 30, teniendo así una diferencia de 20 registros. Por lo tanto, en el script de Sqoop se le pasa como argumento, que el máximo identificador en la tabla de *Hadoop* es 10. Sqoop entonces, realiza la consulta a My*SQL* en donde obtiene como respuesta que el máximo identificador es de 30, por consiguiente, realiza una importación solicitando todos los tuits con id mayor a 10 y menores a 30.

Sincronizar las tablas, implica que se realice una consulta por cada tabla de *Hadoop* que se quiera sincronizar, para obtener el valor máximo de los identificadores y agregarlos en nuestros comandos para realizar la sincronización.

El proceso de sincronización se requiere que sea de manera automática, por lo que se tendrá un programa que contiene la clase Maker como lo muestra la Ilustración 47, el cual tiene como objetivo

generar el comando que ejecutará *Apache* Sqoop, para la sincronización de los datos entre el esquema relacional y el esquema de Hadoop.

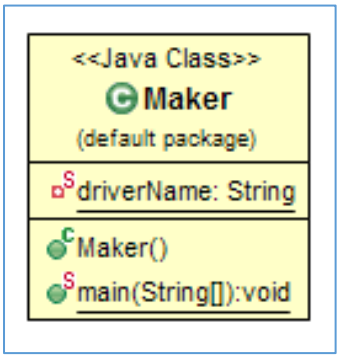


Ilustración 47. Clase que genera la sentencia para el ETL de Sqoop.

El programa utiliza un archivo con propiedades para facilitar la configuración del programa en su ejecución, que contiene los siguientes parámetros:

```
url = jdbc:mySQL://192.168.0.5/social_network
HiveUrl=jdbc\:Hive2\://172.17.0.2\:10000
driver = com.mySQL.jdbc.Driver
username = root
password = root
useSSL=false
path= ETL.sh
```

Detallando el código de la Ilustración 48 tenemos que:

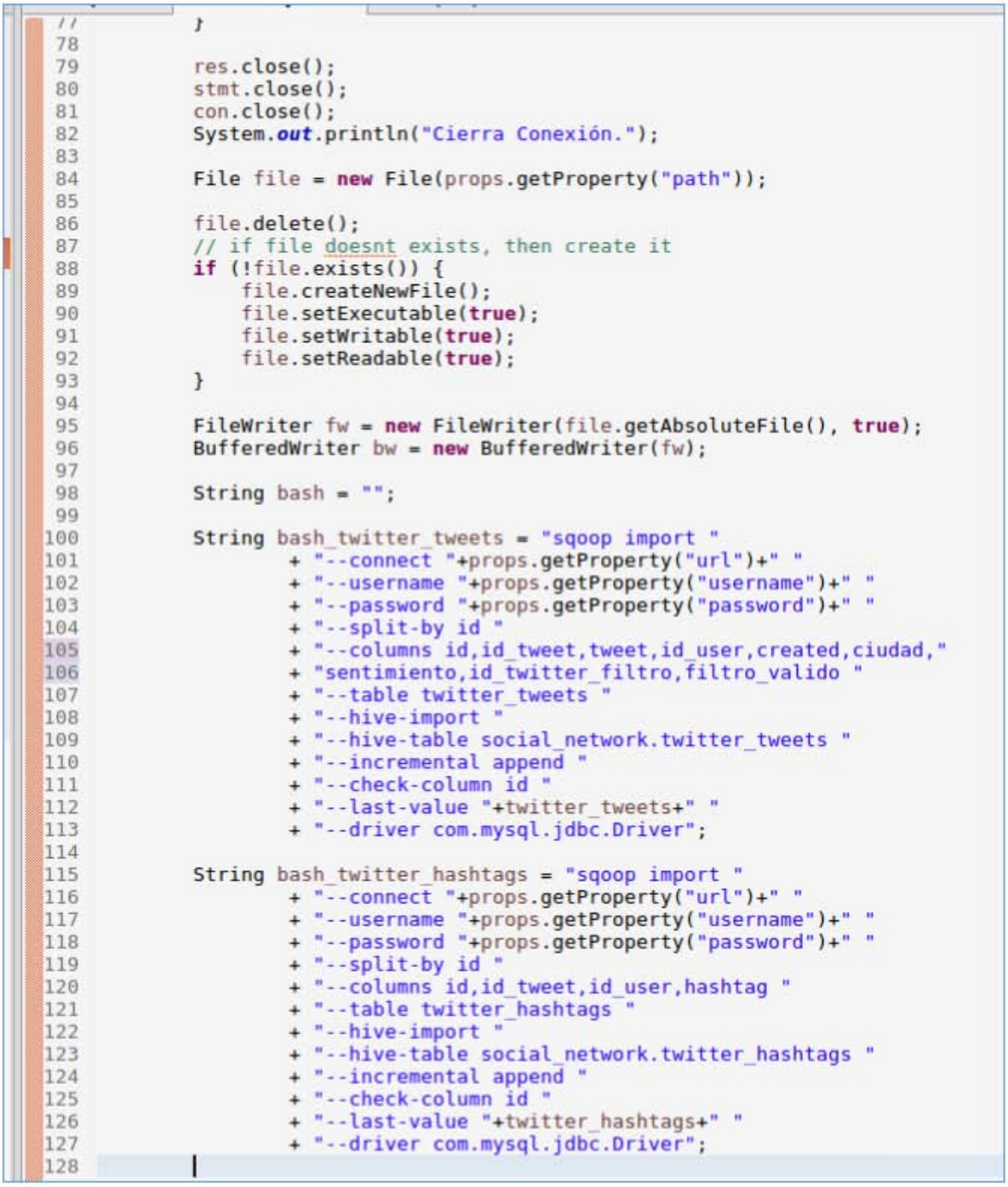
1. En la línea 23 se define el conector de *Hive* para realizar las consultas a la base de datos de Hadoop.
2. En la línea 28 y 29 se declaran las variables que contendrán los máximos identificadores de las cuatro tablas que vamos a sincronizar.
3. En la línea 36 se asigna el driver de *Hive*.
4. En la línea 42 se realiza la conexión a *Hive* mediante la url contenida en la propiedad *Hive*Url.
5. En la línea 48 se crea la sentencia en *HiveQL* para consultar los máximos identificadores en las tablas, uniendo los resultados para iterarlos y almacenarlos en las variables.
6. En la línea 63 se iteran los resultados para almacenarlos en las variables que van a generar el comando.

```java
public class Maker {

    private static String driverName = "org.apache.hive.jdbc.HiveDriver";

    public static void main(String[] args) throws IOException, SQLException {
        System.out.println("Inicia ETL.");

        int twitter_hashtags = 0;       int twitter_tweets = 0;
        int twitter_tweets_filtro_palabra = 0;      int twitter_user = 0;

        Properties props = new Properties();        FileInputStream fis = null;
        fis = new FileInputStream("db.properties");
        props.load(fis);

        try {
            Class.forName(driverName);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            System.exit(1);
        }
        System.out.println("Inicia Conexión.");
        Connection con = DriverManager.getConnection(
                props.getProperty("hiveUrl")+"/social_network");

        Statement stmt = con.createStatement();
        System.out.println("Conexión Creada.");

        String sql;
        sql = "select 'twitter_tweets' as tabla, max(id) from twitter_tweets union"
        + " select 'twitter_hashtags' as tabla, max(id) from twitter_hashtags union"
        + " select 'twitter_tweets_filtro_palabra' as tabla, max(id) "
        + "from twitter_tweets_filtro_palabra union"
        + " select 'twitter_user' as tabla, max(id) from twitter_user";
        ResultSet res;

        System.out.println("Prepara Ejecución.");

        // show tables
        System.out.println("Running: " + sql);
        res = stmt.executeQuery(sql);
        System.out.println("Termina Ejecución.");

        while (res.next()) {
            if (res.getString(1).equals("twitter_hashtags")) {
                twitter_hashtags = Integer.parseInt( res.getString(2) );
            }
            if (res.getString(1).equals("twitter_tweets")) {
                twitter_tweets = Integer.parseInt( res.getString(2) );
            }
            if (res.getString(1).equals("twitter_tweets_filtro_palabra")) {
                twitter_tweets_filtro_palabra = Integer.parseInt( res.getString(2) );
            }
            if (res.getString(1).equals("twitter_user")) {
                twitter_user = Integer.parseInt( res.getString(2) );
            }
            System.out.println(res.getString(1)+" - "+res.getString(2));
        }
```

Ilustración 48. Código que genera script para Sqoop (1/3)

En la Ilustración 49 tenemos que:

7. En la línea 84 se define el path del archivo de configuración del programa, que contendrá el comando generado para sincronizar las tablas.
8. En la línea 100 y 115 se genera el comando para sincronizar la tabla twitter_tweets y twitter_*hashtags* respectivamente.



```
78
79          res.close();
80          stmt.close();
81          con.close();
82          System.out.println("Cierra Conexión.");
83
84          File file = new File(props.getProperty("path"));
85
86          file.delete();
87          // if file doesnt exists, then create it
88          if (!file.exists()) {
89              file.createNewFile();
90              file.setExecutable(true);
91              file.setWritable(true);
92              file.setReadable(true);
93          }
94
95          FileWriter fw = new FileWriter(file.getAbsoluteFile(), true);
96          BufferedWriter bw = new BufferedWriter(fw);
97
98          String bash = "";
99
100         String bash_twitter_tweets = "sqoop import "
101                 + "--connect "+props.getProperty("url")+" "
102                 + "--username "+props.getProperty("username")+" "
103                 + "--password "+props.getProperty("password")+" "
104                 + "--split-by id "
105                 + "--columns id,id_tweet,tweet,id_user,created,ciudad,"
106                 + "sentimiento,id_twitter_filtro,filtro_valido "
107                 + "--table twitter_tweets "
108                 + "--hive-import "
109                 + "--hive-table social_network.twitter_tweets "
110                 + "--incremental append "
111                 + "--check-column id "
112                 + "--last-value "+twitter_tweets+" "
113                 + "--driver com.mysql.jdbc.Driver";
114
115         String bash_twitter_hashtags = "sqoop import "
116                 + "--connect "+props.getProperty("url")+" "
117                 + "--username "+props.getProperty("username")+" "
118                 + "--password "+props.getProperty("password")+" "
119                 + "--split-by id "
120                 + "--columns id,id_tweet,id_user,hashtag "
121                 + "--table twitter_hashtags "
122                 + "--hive-import "
123                 + "--hive-table social_network.twitter_hashtags "
124                 + "--incremental append "
125                 + "--check-column id "
126                 + "--last-value "+twitter_hashtags+" "
127                 + "--driver com.mysql.jdbc.Driver";
128
```

Ilustración 49. Código que genera script para Sqoop (2/3)

En la Ilustración 50 tenemos que:

9. En las líneas 129 y 143 se genera el comando para sincronizar la tabla twitter_tweets y twitter_*hashtags* respectivamente.
10. En la línea 159 se juntan los comandos generados concatenados con *&&* que reconoce Linux como un conector secuencial, esto para que se ejecute uno detrás de otro.
11. En la línea 164 se guarda el comando generado en el archivo definido con extensión *.sh,* para ejecutarlo desde la terminal de Linux.

```
128
129         String bash_twitter_tweets_filtro_palabra = "sqoop import "
130                 + "--connect "+props.getProperty("url")+" "
131                 + "--username "+props.getProperty("username")+" "
132                 + "--password "+props.getProperty("password")+" "
133                 + "--split-by id "
134                 + "--columns id,id_filtro_palabra,id_tweet "
135                 + "--table twitter_tweets_filtro_palabra "
136                 + "--hive-import "
137                 + "--hive-table social_network.twitter_tweets_filtro_palabra "
138                 + "--incremental append "
139                 + "--check-column id "
140                 + "--last-value "+twitter_tweets_filtro_palabra+" "
141                 + "--driver com.mysql.jdbc.Driver";
142
143         String bash_twitter_user = "sqoop import "
144                 + "--connect "+props.getProperty("url")+" "
145                 + "--username "+props.getProperty("username")+" "
146                 + "--password "+props.getProperty("password")+" "
147                 + "--split-by id "
148                 + "--columns id,id_user,screen_name,real_name,description,"
149                 + "friends_count,followers_count,location "
150                 + "--table twitter_user "
151                 + "--hive-import "
152                 + "--hive-table social_network.twitter_user "
153                 + "--incremental append "
154                 + "--check-column id "
155                 + "--last-value "+twitter_user+" "
156                 + "--driver com.mysql.jdbc.Driver";
157
158
159         bash = bash_twitter_hashtags + " && "
160                 + bash_twitter_tweets + " && "
161                 + bash_twitter_tweets_filtro_palabra + " && "
162                 + bash_twitter_user;
163
164         bw.write(bash);
165         bw.write(System.getProperty("line.separator"));
166         bw.close();
167         System.out.println("Archivo etl.sh generado.");
168
169
```

Ilustración 50. Código que genera script para Sqoop (3/3)

Al ejecutar el programa, se tendrá la salida que se muestra en la Ilustración 51, en donde se puede ver que se muestra el *query* en *Hive*QL, para obtener los máximos identificadores de las tablas de la base da datos de Hadoop, para posteriormente imprimirlos y generar el archivo *ETL.sh* que contiene el comando.

Ilustración 51. Salida del programa que genera el script para sincronizar los datos utilizando Sqoop.

El contenido del archivo *ETL.sh* se muestra en Ilustración 52.

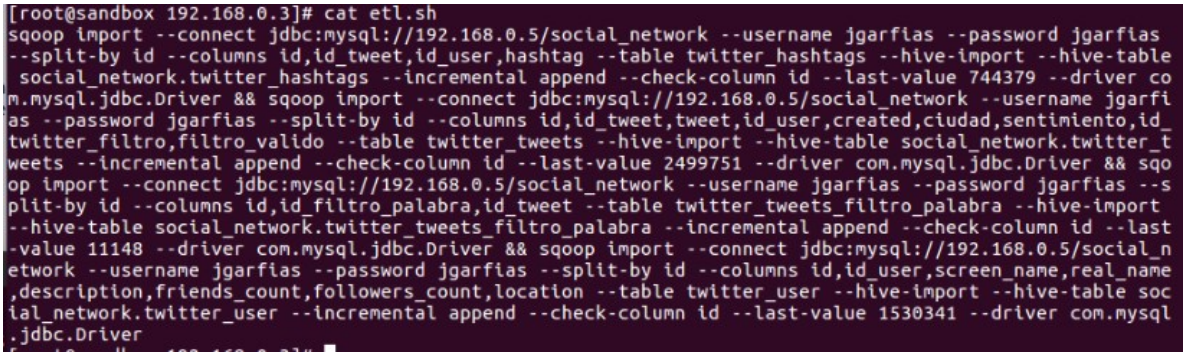


Ilustración 52. Contenido del archivo ETL.sh con el script para ejecutar la sincronización de datos con Sqoop.

Al ejecutar nuestro archivo *ETL*.sh, se muestra la salida de la terminal en la Ilustración 53, en donde se puede ver al final de la imagen, el máximo identificador contenido en la tabla de *Hadoop* de 744,379, y el valor de la tabla de My*SQL* de 978,424.

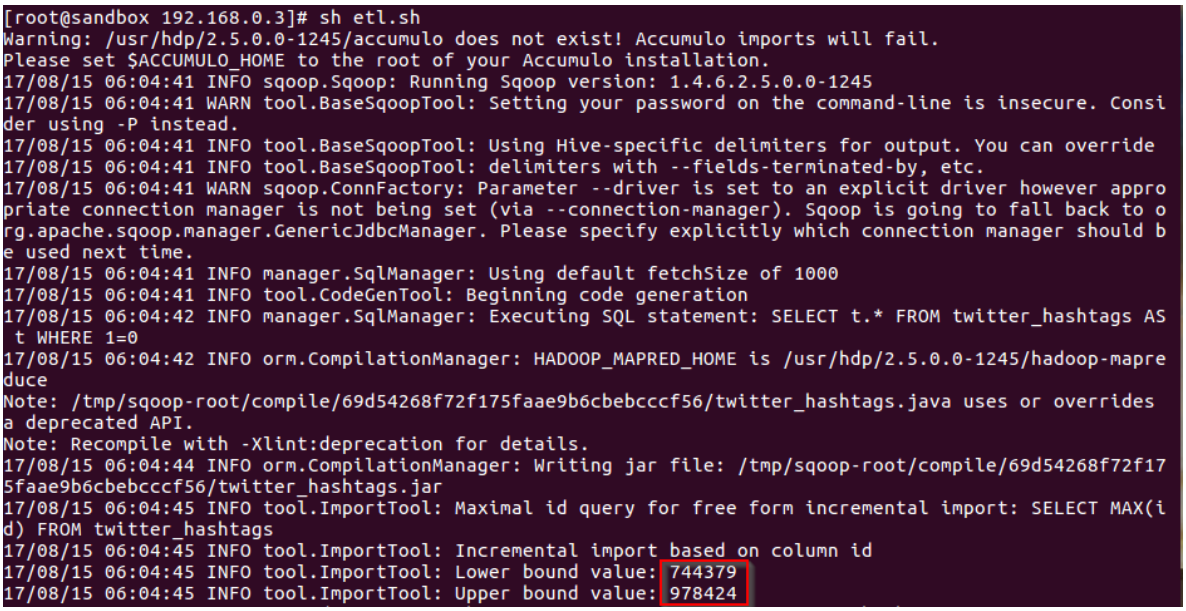


Ilustración 53. Ejecución del archivo ETL.sh para la sincronización de datos (1/2).

En la Ilustración 54 se puede observar como en la secuencia de la ejecución de Sqoop, obtiene el mínimo y el máximo de los identificadores para posteriormente hacer la división entre cuatro, para realizar la consulta en paralelo y agregarlos a la base de datos en Hadoop.

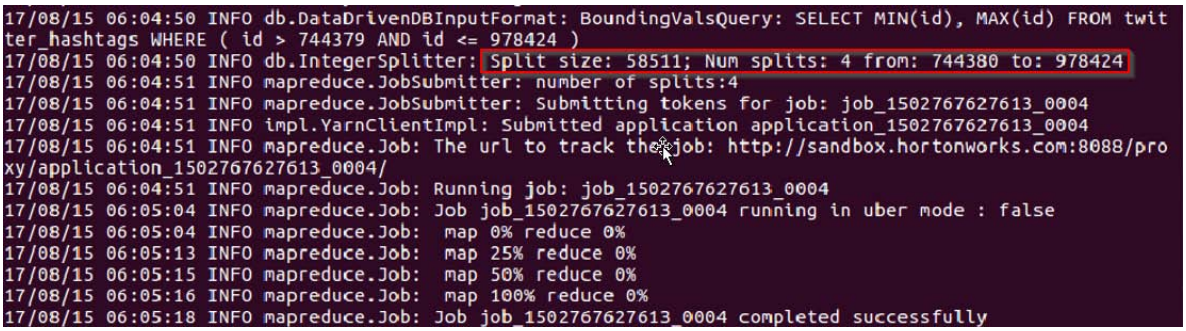


Ilustración 54. Ejecución del archivo ETL.sh para la sincronización de datos (2/2).

Una vez terminados los procesos de sincronización, se puede observar en la Ilustración 55 y la Ilustración 56 que nuestros máximos identificadores, son los mismos para *Hadoop* como para My*SQL*.

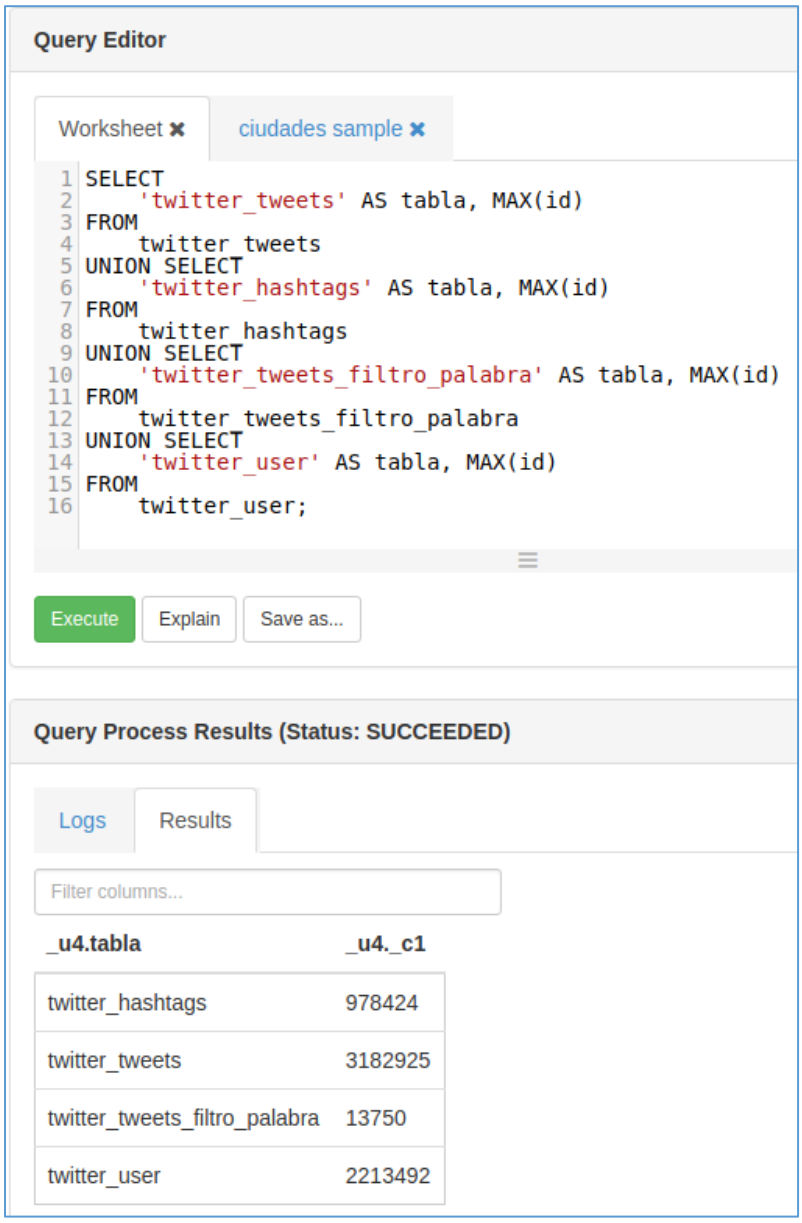


Ilustración 55. Consulta de máximos identificadores de las tablas en Hadoop con HiveQL.

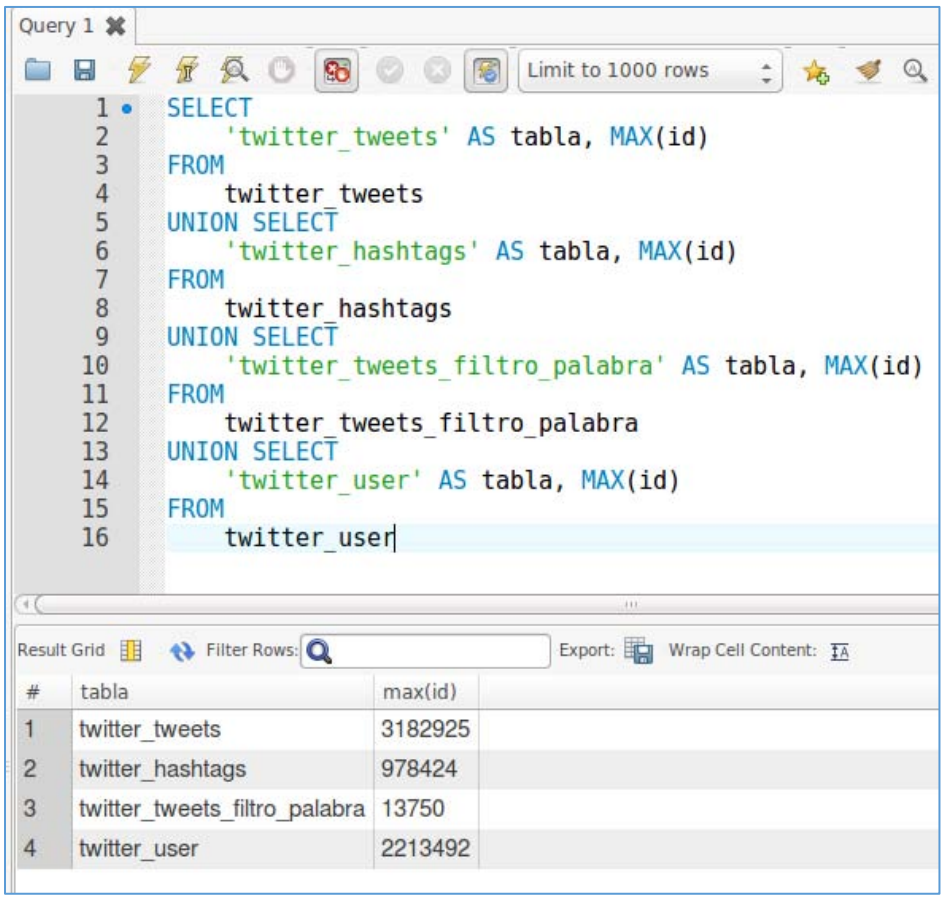


Ilustración 56. Consulta de máximos identificadores de las tablas en MySQL.

4.4 Herramientas para el análisis de *Big Data*

Antes de pasar al análisis de *Big Data*, se deben de conocer las herramientas y su operación, para comprender su integración en el proyecto, por lo que se explicará el uso de *Apache Hive* para realizar las consultas a los archivos HDF de Hadoop, *Pig Latin* para la programación de *ETL* sobre los HDF, cuya introducción se realizó en la sección 1.8 y 1.9, y la herramienta de *Apache* Zeppelin, para la graficación automática de los resultados, para entender cómo trabajar con las tres herramientas, y en conjunto generar resultados gráficos.

4.4.1 Uso de *apache Hive*

Como se comentó en el la sección 1.8, *Hive* es una herramienta para acceder a los datos almacenados en Hadoop. Se explicará cómo se crea una base de datos con *Hive*, y su lenguaje *Hive*QL, el cual se verá que tiene mucha similitud con *SQL*.

De igual manera, utilizando la interfaz que nos provee el marco de trabajo de Hortonworks, cómo se muestra en la Ilustración 57, en donde podremos realizar nuestras consultas de una manera más sencilla.

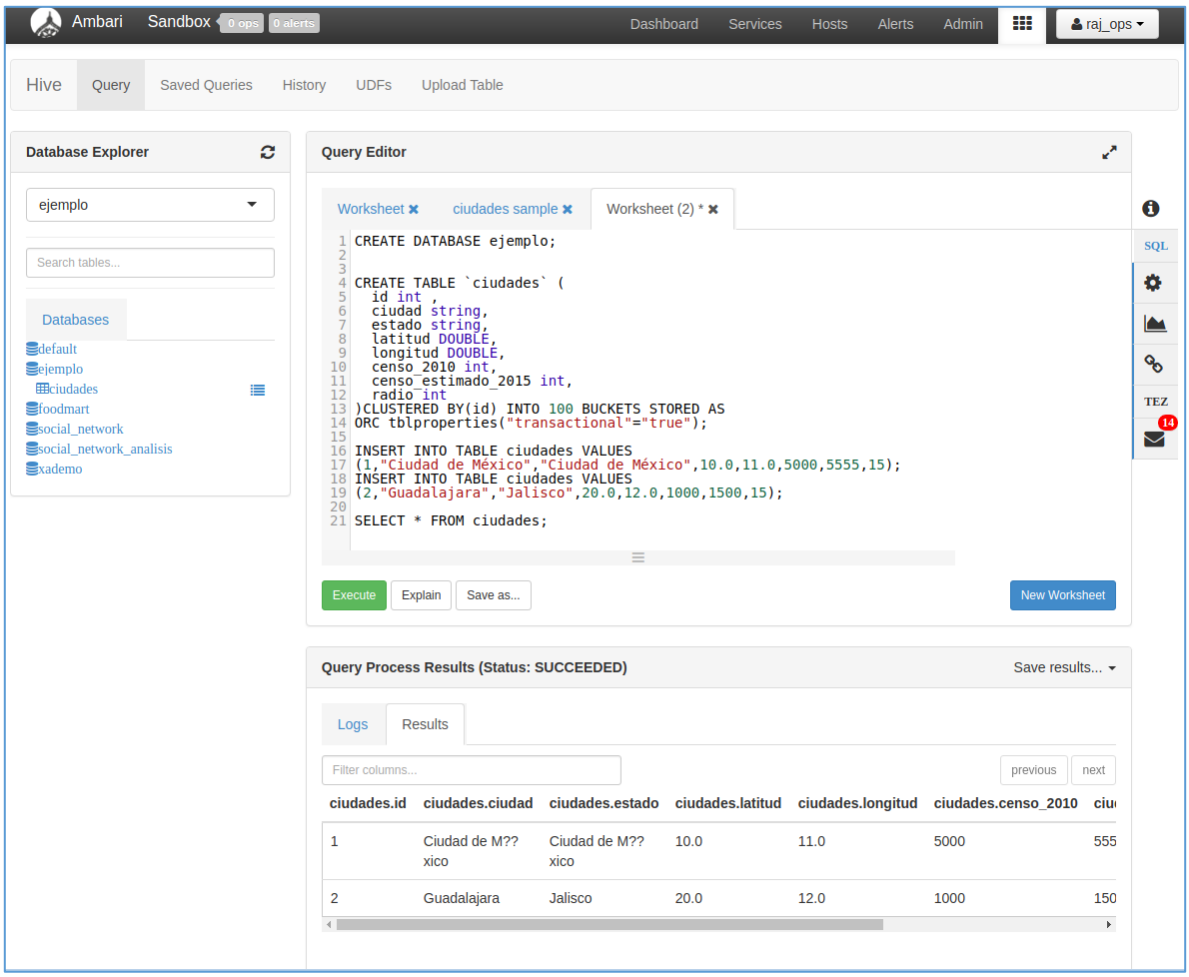


Ilustración 57. Interfaz de Hive que provee Hortonworks.

En la Ilustración 58 se muestran ejemplos de sentencias en lenguaje *Hive*QL:

1. En la línea 1 se muestra una sentencia para crear una base de datos.
2. En la línea 4 se representa una sentencia para crear una tabla llamada ciudades, la cual en el sistema de archivos de Hadoop.
    a. Se especifica que se repartirá en 100 *BUCKETS* los cuales son básicamente carpetas que contienen los archivos de datos, aplicando una función hash sobre el id para determinar en que carpeta se guardará el registro dentro de su archivo.
    b. La propiedad dentro de *tblproperties* que en este caso es *"transactional"="true"*, permite transacciones ACID del tipo insert, update y delete.
3. En la línea 16 y 18 se muestra la sentencia para insertar datos en la tabla ciudades.
4. En la línea 21 se muestra la sentencia para consultar la tabla de ciudades, la cual da como resultado lo que se muestra en Ilustración 59.

```
1  CREATE DATABASE ejemplo;
2
3
4  CREATE TABLE `ciudades` (
5    id int ,
6    ciudad string,
7    estado string,
8    latitud DOUBLE,
9    longitud DOUBLE,
10   censo_2010 int,
11   censo_estimado_2015 int,
12   radio int
13 )CLUSTERED BY(id) INTO 100 BUCKETS STORED AS
14 ORC tblproperties("transactional"="true");
15
16 INSERT INTO TABLE ciudades VALUES
17 (1,"Ciudad de México","Ciudad de México",10.0,11.0,5000,5555,15);
18 INSERT INTO TABLE ciudades VALUES
19 (2,"Guadalajara","Jalisco",20.0,12.0,1000,1500,15);
20
21 SELECT * FROM ciudades;
```

Ilustración 58. Sentencias básicas en HiveQL.

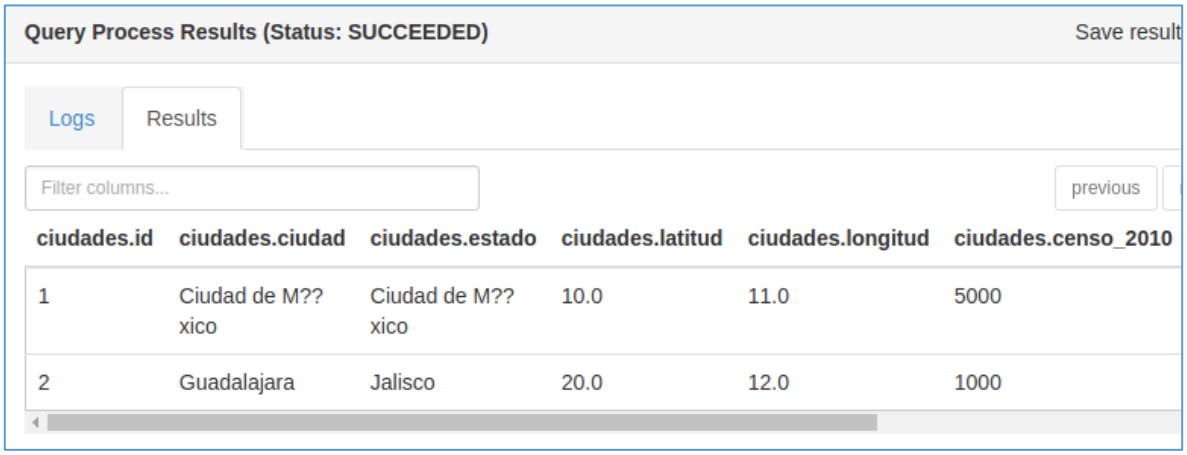


Ilustración 59. Resultado de la consulta a la tabla de ciudades.

En la Ilustración 60, se puede observar la base de datos llamada social_network que se tiene en My*SQL*, ahora almacenada en la base de datos de Hadoop. Consultando su contenido con *HiveQL* sobre la tabla twitter_tweets, se puede ver que contiene los tuits que se consultaron mediante la *API* de twitter.

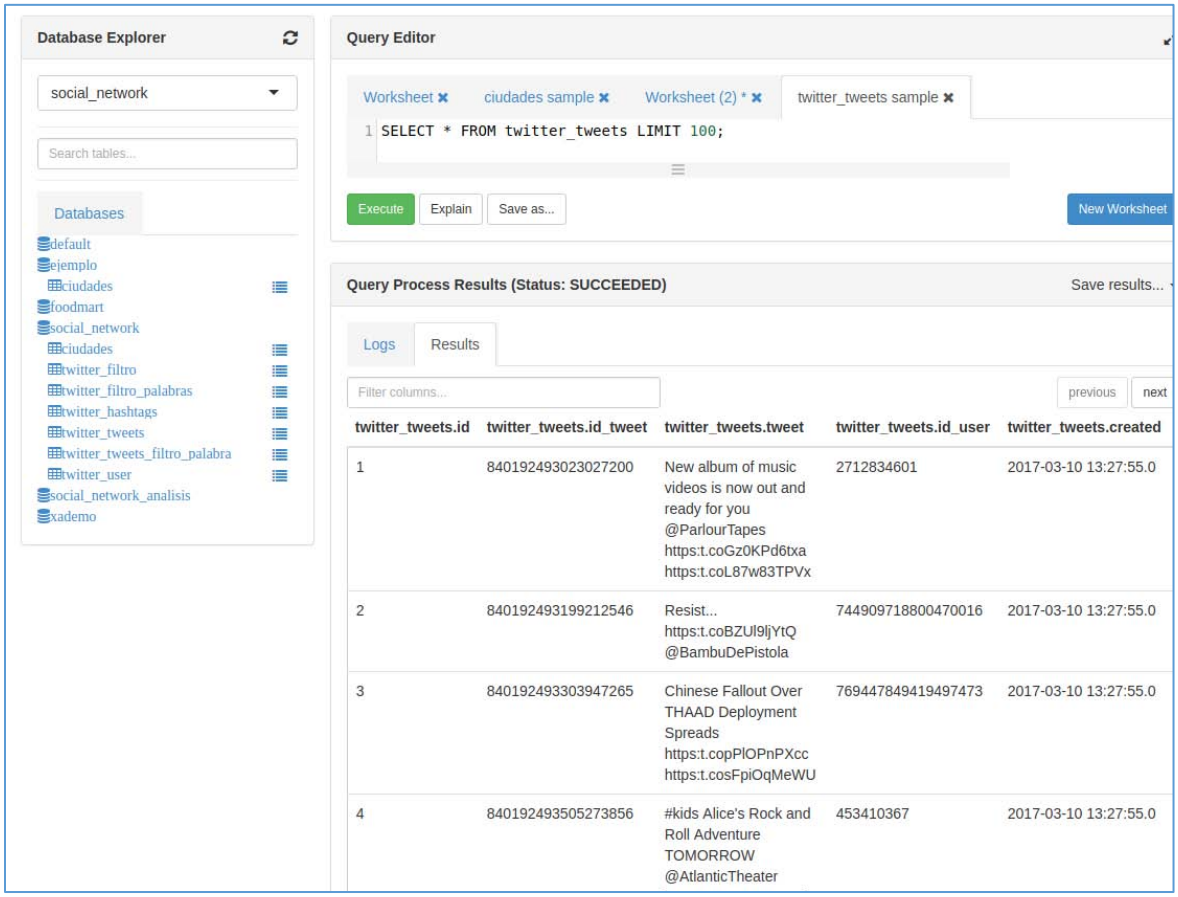


Ilustración 60. Base de datos en Hadoop consultada por HiveQL.

4.4.2 Uso de *Pig Latin*

Como se comentó en la sección 1.9, esta herramienta nos sirve para crear un *ETL* sobre una gran variedad de fuentes de datos y almacenarlos en Hadoop, archivos de texto planos u otra base de datos añadiéndole los conectores adecuados.

De igual manera, Hortonworks provee una interfaz para manipular y crear nuestros propios scripts como se muestra en la Ilustración 61.

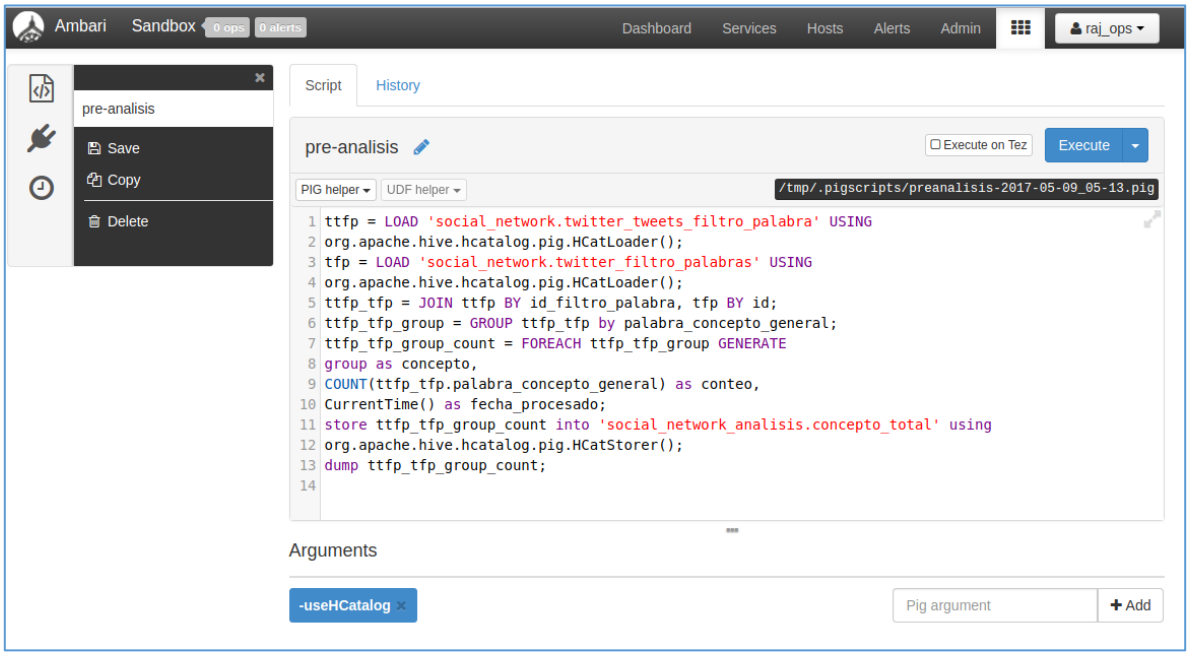


Ilustración 61. Interfaz de Pig Latin en Hortonworks.

Pig Latin cuenta con una gran variedad de sentencias para relacionar datos, por lo que se explicarán a continuación los utilizados en la Ilustración 62:

1. En la línea 1, el contenido de la tabla **twitter_tweets_filtro_palabra** de la base de datos **social_network,** se almacena en un contenedor llamado **ttfp** utilizando un conector de *Hive* para realizar la carga de los datos.
2. En la línea 3, el contenido de la tabla **twitter_filtro_palabras** de la base de datos **social_network,** se almacena en un contenedor llamado **tfp** utilizando un conector de *Hive* para realizar la carga de los datos.
3. En la línea 5, se relacionan ambas tablas indicando la columna de referencia de cada una, y el resultado se almacena en un contenedor llamado **ttfp_tfp**. Este comando se comporta como un JOIN en *SQL*.
4. En la línea 6, se agrupa la columna **palabra_concepto_general** contenido en **ttfp_tfp**, y almacena el resultado en **ttfp_tfp_group**.
5. En la línea 7, se genera la tabla con el resultado del comando, en donde se le indica que por cada registro de la agrupación contenida en **ttfp_tfp_group,** genera otra agrupación sobre el concepto, y realiza la cuenta del número de apariciones del concepto en el JOIN contenido en **ttfp_tfp** llamándola conteo, y le añade una fecha en la que se ejecutó el comando.
6. En la línea 11, almacena el resultado de la consulta contenida en **ttfp_tfp_group_count** en la tabla **concepto_total** de la base de datos **social_network_analisis** utilizando el conector de *Hive*.
7. En la línea 13, imprime el resultado en la consola.

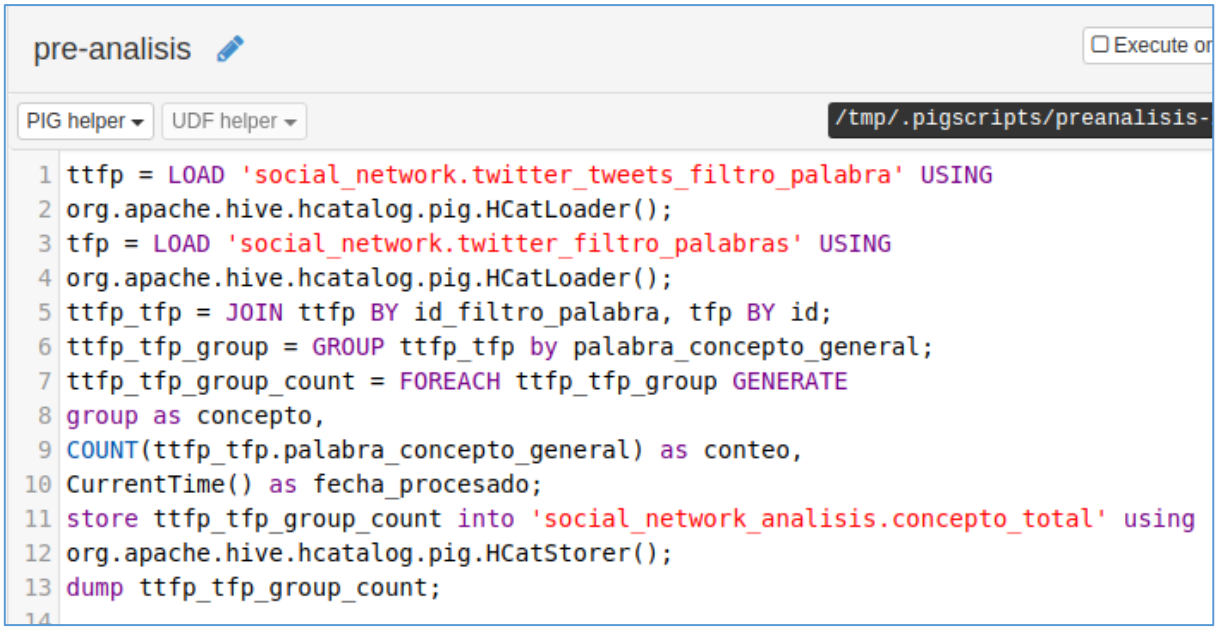


Ilustración 62. Script en Pig Latin.

En la Ilustración 63, se puede ver el resultado de la ejecución del script. El cuál es el total de tuits con menciones referentes a las drogas definidas en el catálogo de palabras, las cuales están en el texto de los tuits.

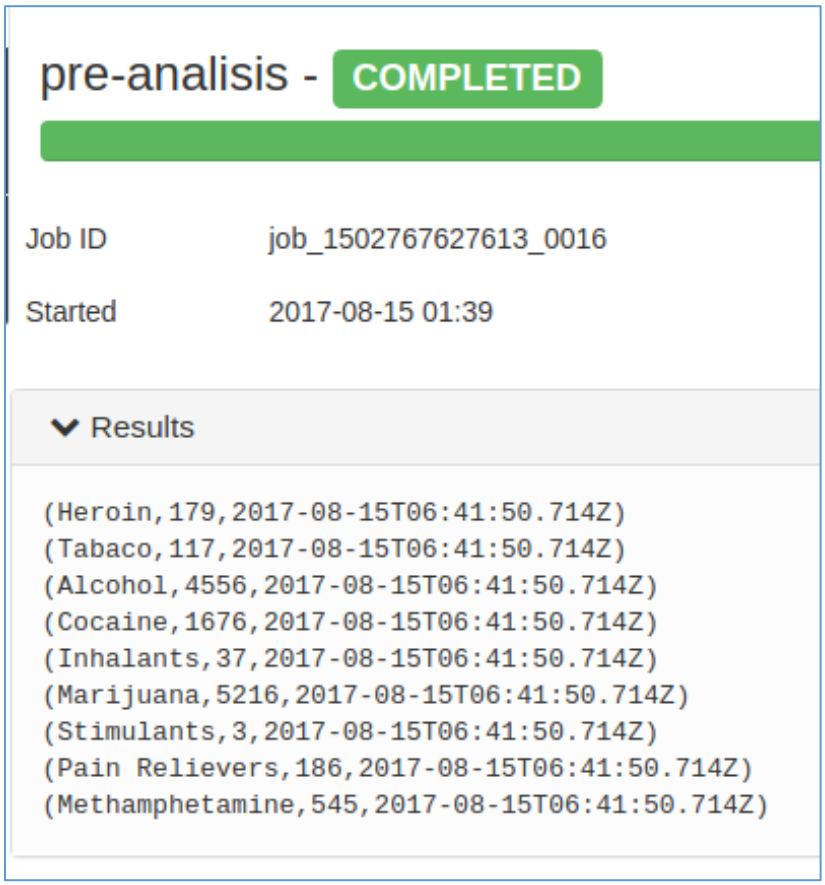


Ilustración 63. Resultado de la ejecución del script en Pig Latin.

En la Ilustración 64 se puede observar el contenido de la tabla *concepto_total* que se llenó mediante el script de *Pig Latin*.

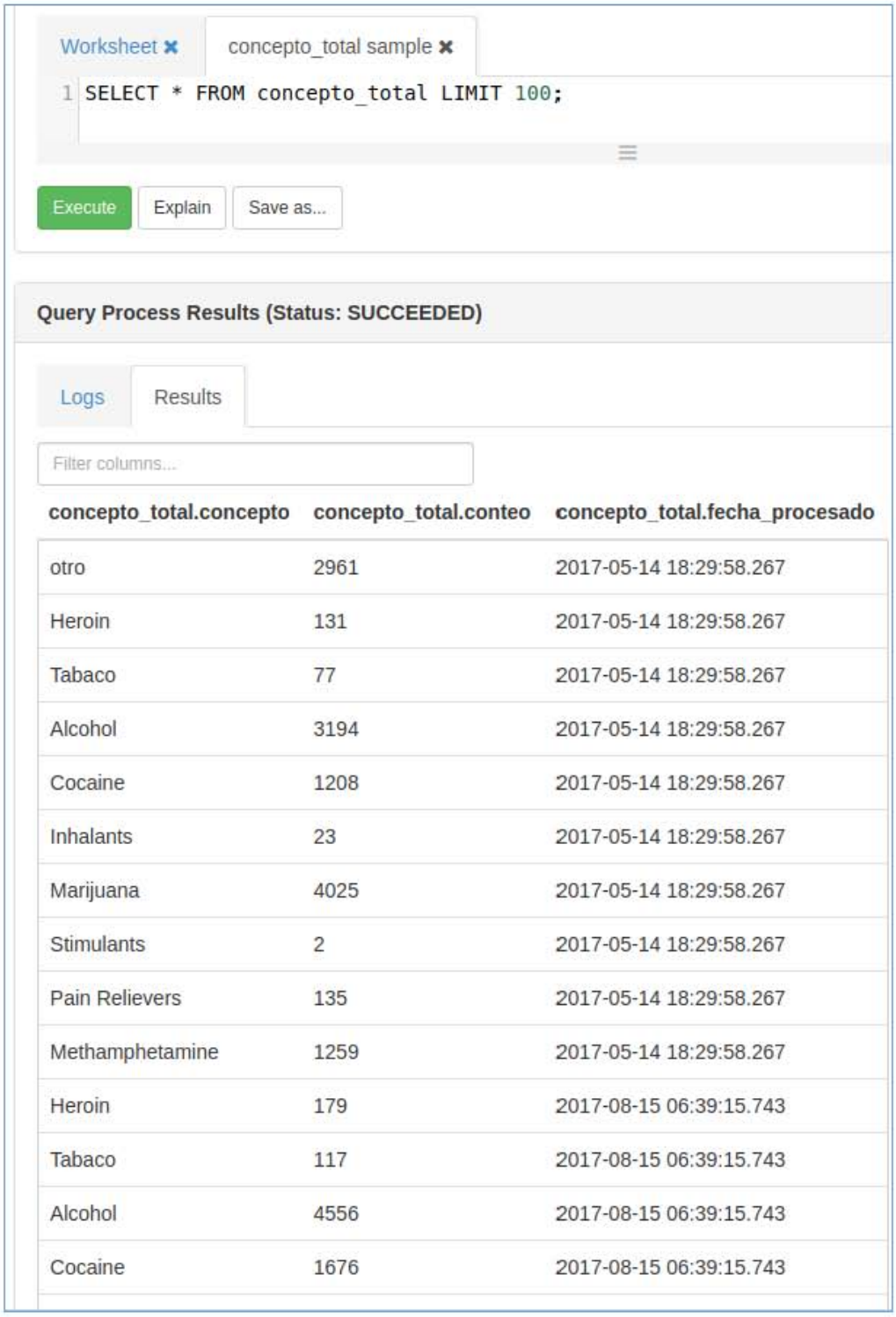


Ilustración 64. Consulta de la tabla generada por el script en Pig Latin en la interfaz de Hive.

4.4.3 Uso de Zeppelin

La herramienta de Zeppelin, como se puede ver su interfaz de inicio en la Ilustración 65, es una herramienta que nos facilitará realizar el análisis de los datos de una manera gráfica, utilizando lenguaje *HiveQL* y seleccionando de una gran variedad de estilos de gráficas.

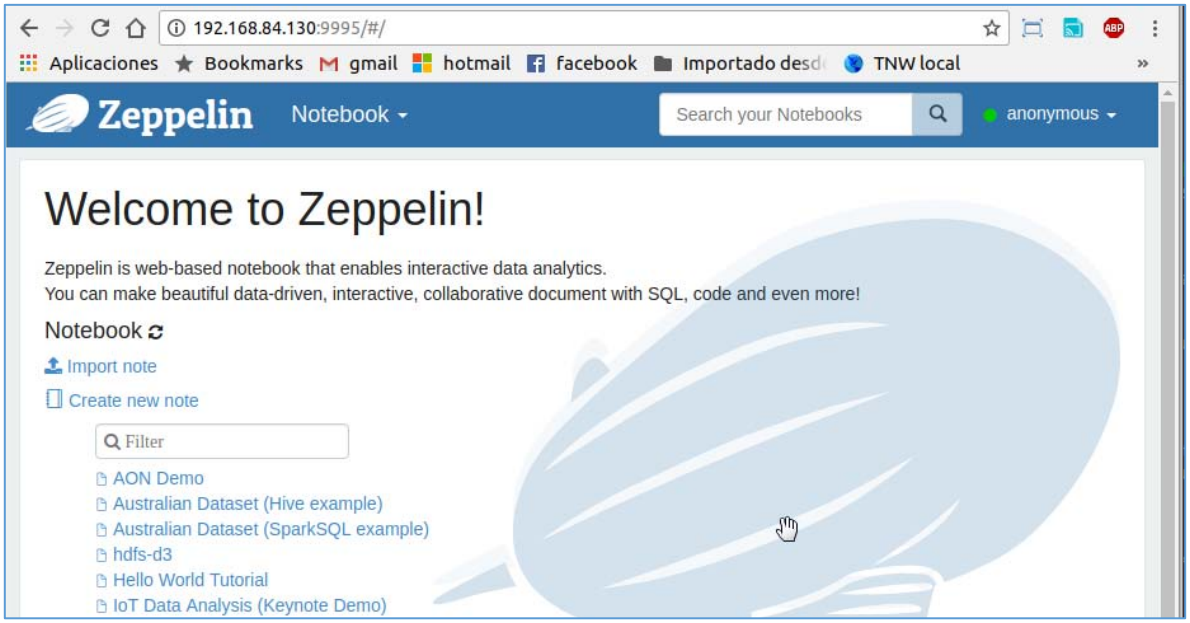


Ilustración 65. Interfaz de inicio de Zeppelin.

Se realiza una consulta a los datos almacenados en la tabla de *concepto_total*, para obtener el resultado que se muestra en la Ilustración 66.

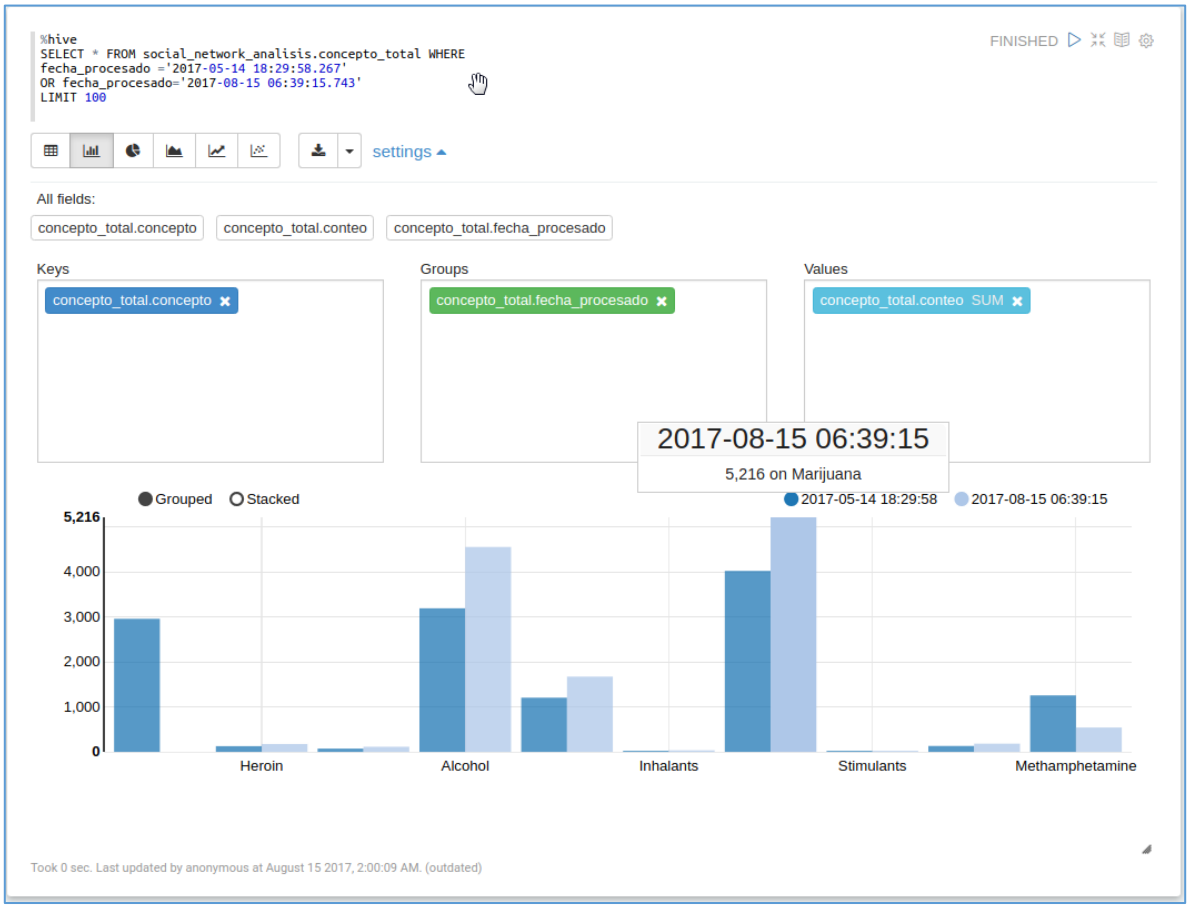


Ilustración 66. Gráfica generada por resultados del query en HiveQL y graficada por Zeppelin.

El query de la Ilustración 67, realiza la consulta sobre dos fechas definidas.

```
%hive
SELECT * FROM social_network_analisis.concepto_total WHERE
fecha_procesado ='2017-05-14 18:29:58.267'
OR fecha_procesado='2017-08-15 06:39:15.743'
LIMIT 100
```

Ilustración 67. Query en HiveQL que consulta los resultados en dos fechas definidas.

Se genera la gráfica de la Ilustración 68, en donde se observa el aumento en el número de menciones capturadas el 14 de Mayo de 2017 y al 15 de Agosto de 2017.

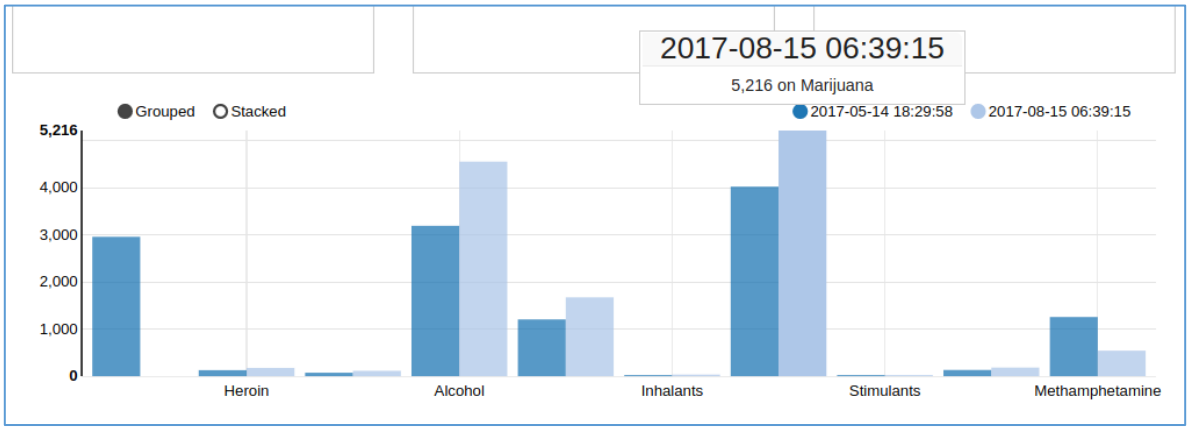


Ilustración 68. Gráfica del query en HiveQL.

En la Ilustración 69 se puede ver la configuración de valores en donde se definen sus *Keys* o eje X, sus *Groups* o agrupación, y sus *Values* o valores.

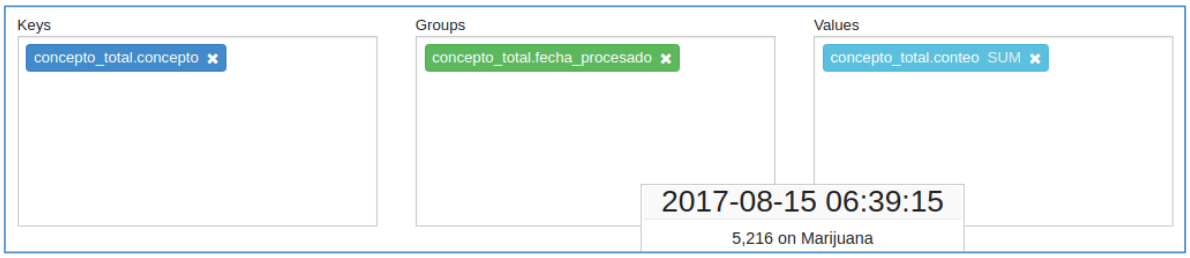


Ilustración 69. Opciones de manipulación de datos de Zeppelin.

Capítulo 5. Agregación, análisis y visualización de resultados

En este capítulo se abordarán las etapas de Agregación de los Datos y Representación en donde se justifica la definición de los grupos de palabras que contienen las drogas, la etapa de Análisis de los Datos en donde se define de qué contenedores de datos se podrán responder las preguntas, y la etapa de Visualización de los Datos, en donde se define la manera ideal de representar gráficamente los resultados para su mejor entendimiento.

Tomando los datos ya generados y almacenados en la base de datos de Hadoop, se procederá a realizar las consultas en *HiveQL* para obtener los resultados que contestarán las preguntas planteadas en el Capítulo 3 en la etapa de la Evaluación del Caso de Negocio.

5.1 Agregación de los datos y representación

En el proceso de agregación, se realizó un estudio previo para identificar los sinónimos de las palabras que serán buscadas en el texto de un tuit, ya que en el lenguaje coloquial, existen muchas maneras de referirse a una cosa. El ejemplo más claro en el dominio del problema abordado, es la mariguana, la cual tiene un gran número de sinónimos en inglés, como por ejemplo *weed, kush, cannabis, ganja, bhang, hashish, mojo*, etc. Por lo tanto para realizar una búsqueda lo más precisa posible, se deben de agregar el mayor número de sinónimos conocidos posible para buscar en la cadena de texto que conforma el tuit, y agruparla en su nombre raíz de la droga.

Se agregó como droga palabras relacionadas con el alcohol y el tabaco, las cuales no aborda la encuesta nacional de EUA, esto para que podamos relacionar la aparición de drogas ilegales combinadas con drogas legales como se verá más adelante.

Tabla 10. Sinónimos de las drogas a buscar en los tuits con su droga raíz.

Palabra	Droga
beer	Alcohol
alcohol	Alcohol
cerveza	Alcohol
vodka	Alcohol
tequila	Alcohol
whisky	Alcohol
whiski	Alcohol
whyski	Alcohol
whiskey	Alcohol
cocaine	Cocaine
Coke	Cocaine
Crack	Cocaine
happy dust	Cocaine
nose Candy	Cocaine
Stardust	Cocaine

white horse	Cocaine
Palabra	**Droga**
white lady	Cocaine
angel dust	Cocaine
heroin	Heroin
***APL*ates**	Heroin
snort	Inhalants
weed	Marijuana
marijuana	Marijuana
kush	Marijuana
cannabis	Marijuana
mariguana	Marijuana
canabis	Marijuana
ganja	Marijuana
bhang	Marijuana
hashish	Marijuana
mojo	Marijuana
Methamphetamine	Methamphetamine
meth	Methamphetamine
crank	Methamphetamine
methadrine	Methamphetamine
aspirin	Pain Relievers
morphine	Pain Relievers
opioids	Pain Relievers
fentanyl	Pain Relievers
hydrocodone	Pain Relievers
methadone	Pain Relievers
opiate	Pain Relievers
opium	Pain Relievers
barbiturates	Sedatives
amphetamine	Stimulants
big h	Stimulants
analeptic	Stimulants
tabaco	Tabaco
cigar	Tabaco

5.2 Análisis y visualización de los Datos

En la etapa de análisis, se utilizarán los datos obtenidos y se tomará en cuenta principalmente, cómo se relacionan las tablas importadas en el ambiente de trabajo de Hortonworks, ya que se efectuarán relaciones entre ellas, para poder obtener resultados sobre las preguntas planteadas en la etapa de la evaluación del caso de negocio.

Por otra parte, en la etapa de visualización de los datos, se debe de buscar la manera de expresar los resultados lo más clara posible, eligiendo una representación gráfica adecuada de los datos que se quieran exponer como se verá a continuación, en donde por cada pregunta se plantea el análisis y su gráfica de resultados.

5.3 De acuerdo a los resultados de las encuestas, ¿Existe una presencia en twitter similar a las 10 drogas más consumidas reportadas por la encuesta de la NSDUH?

Análisis

El análisis de frecuencia de las menciones la vamos a comparar con el resultado de la encuesta realizada por la NSDUH, con la cual sabremos si hay una presencia similar en las redes sociales.

Recordando que se realizó un *ETL* en la sección 4.4.2 sobre *Pig Latin*, en donde se obtuvo el total de menciones sobre las drogas que aparecieron en los tuits, por lo tanto, se realizará una consulta a la tabla concepto_total de la base de datos social_network_analisis, en donde se filtrará por la fecha más reciente, y ordenando el resultado del conteo de registros como se muestra en el Query 1 siguiente:

```
Query 1:
SELECT * FROM
    social_network_analisis.concepto_total
WHERE fecha_procesado IN (
    SELECT max(fecha_procesado) FROM
    social_network_analisis.concepto_total
) ORDER BY conteo
```

El Query 2 sirve para conocer el volumen de menciones, y su distribución en los estados en los que se obtuvieron los tuits. Para lograr esto, se realiza la consulta de todos aquellos tuits que tengan palabras contenidas en el filtro, para después realizar una agrupación de la ciudad y un conteo de los tuits recopilados.

```
Query 2:
SELECT
    ciudad, COUNT(*) conteo
FROM
    (SELECT
        tt.ciudad AS ciudad,
            tt.id_tweet
```

```
          FROM
              social_network.twitter_tweets tt
          INNER JOIN social_network.twitter_tweets_filtro_palabra
      ttfp ON tt.id_tweet = ttfp.id_tweet) tabla
GROUP BY ciudad order by conteo
```

Visualización

Representando el resultado del Query 1, comparándolo con las gráficas de la NSDUH y los propios, que se muestran en la Ilustración 70, se observa lo siguiente:

- La gráfica en donde se muestra que la mariguana está en el primer lugar de la encuesta por parte de la NSDUH, coincide con el el mayor número de menciones en twitter.
- Palabras relacionadas al alcohol que se añadieron en la búsqueda, aparece como segundo lugar en apariciones.
- Los medicamentos para el dolor que aparecen en el segundo lugar de la NSDUH, en los resultados propios están en quinto lugar.
- La cocaína sin embargo, aparece en el tercer puesto en ambos resultados.
- Alucinógenos, sedantes y tranquilizantes no tuvieron presencia en los resultados obtenidos de twitter.
- Los estimulantes tuvieron una gran diferencia, ya que en la NSDUH están en quinto lugar, y en los resultados del análisis están en el noveno puesto.

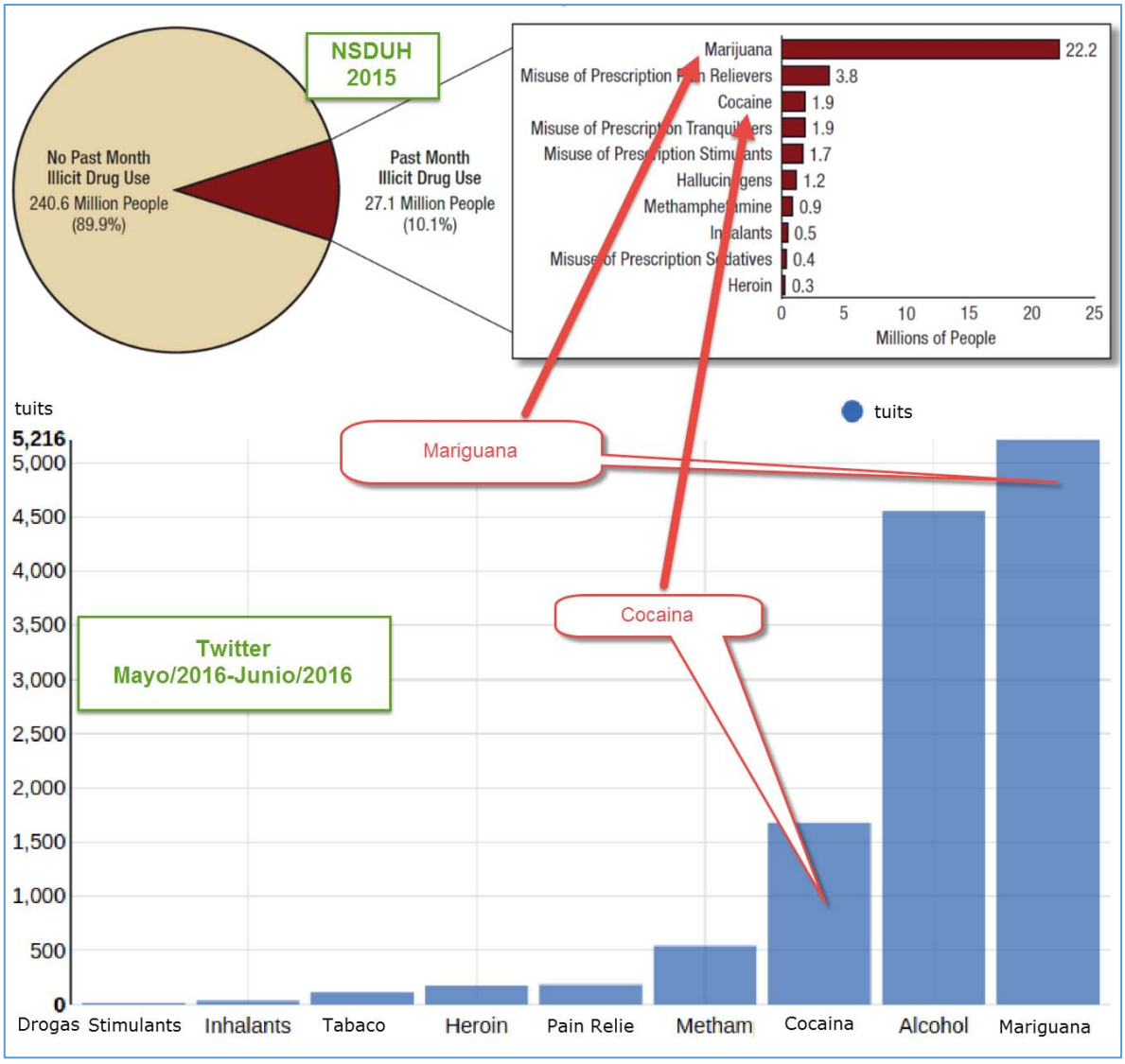


Ilustración 70. Comparación entre resultados de la NSDUH y los resultados obtenidos por el análisis de Big Data.

Basándose en los resultados visualizados, se puede concluir que sí hay relación entre las drogas más conocidas con su presencia en twitter.

En la Ilustración 71, se muestran los estados de las ciudades a las que se les realizó la consulta de tuits.

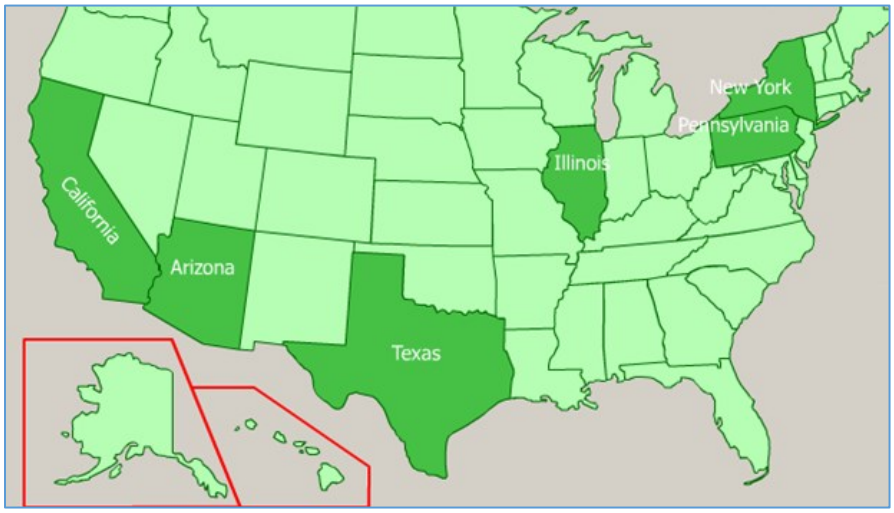


Ilustración 71. Estados de los que se obtuvieron los tuits.

La distribución de las menciones en las ciudades, se muestra en la Ilustración 72, en donde podemos ver lo siguiente:

- San Diego, California, concentra el mayor número de menciones, San Jose, California está en el tercer puesto y Los Angeles, California, está en el décimo puesto.
- Por otra parte, agrupando por estados, obtenemos la Tabla 11, en la que podemos ver que, en el estado de California se presenta el mayor número de menciones y en el estado de Nueva York el menor.

Tabla 11. Número de menciones por estado de E.U.A.

Estado	Número de menciones relacionadas con drogas
California	3,788
Texas	3,199
Arizona	1,412
Pennsylvania	1,112
Illinois	1,045
New York	658

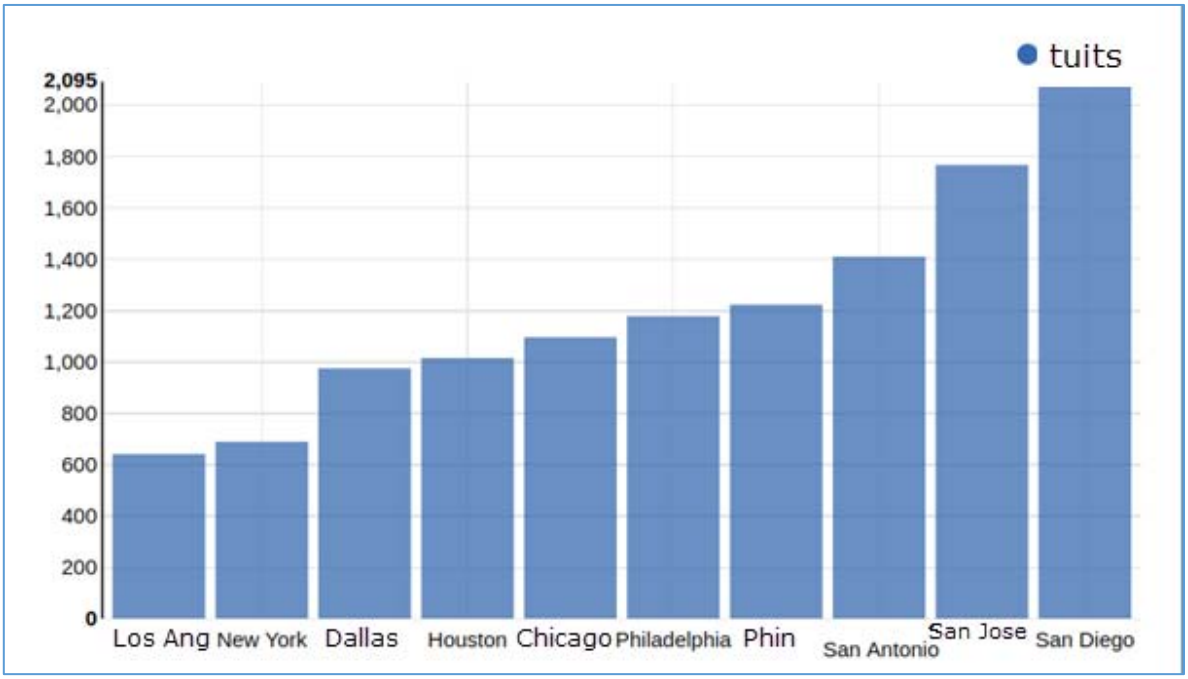


Ilustración 72. Distribución de menciones entre las 10 ciudades más pobladas de EUA.

5.4 ¿Se podrán detectar picos en el consumo de drogas monitoreando twitter en tiempo real?

Análisis

Para obtener la respuesta a esta pregunta, se debe recordar que la aplicación se ejecutó por 31 días seguidos las 24 horas, para poder observar el comportamiento de la aparición de las menciones en el tiempo, detectando que el día 21 de mayo de 2017 se elevó significativamente su número.

El Query 3 relaciona la tabla de tuits con la tabla que contiene los identificadores de las palabras encontradas en los tuits, y esta última se relaciona con el catálogo de palabras. Realizando la consulta entre las fechas del 16 de mayo al 16 de junio del 2017. El resultado preliminar es la cadena de caracteres que representa la fecha, cortándola a 10 caracteres de izquierda a derecha, para que posteriormente se pueda agrupar por día y así poder contabilizar las menciones las menciones diarias.

```
Query 3:
SELECT tiempo, count(*) FROM (
    SELECT
        SUBSTR(created,0,10) AS tiempo
    FROM
        social_network.twitter_tweets tt
    INNER JOIN
        social_network.twitter_tweets_filtro_palabra ttfp
            ON tt.id_tweet = ttfp.id_tweet
    INNER JOIN
```

```
        social_network.twitter_filtro_palabras tfp
            ON ttfp.id_filtro_palabra = tfp.id
        WHERE tt.created BETWEEN
            '2017-05-16 00:00:00' AND '2017-06-16 23:59:59') tabla
GROUP BY tiempo
```

El Query 4 realiza un análisis más detallado en el tiempo, centrándose en el fin de semana en el que se detectó el pico y así poder observar la distribución de las menciones en las horas, filtrando la fecha desde el sábado 20 de mayo a las 00:00 horas, hasta el lunes 22 de mayo a las 23:59 horas del 2017.

```
Query 4:
SELECT tiempo, count(*) conteo FROM (
    SELECT
        SUBSTR(created,0,13) as tiempo
    FROM social_network.twitter_tweets tt
    INNER JOIN
        social_network.twitter_tweets_filtro_palabra ttfp ON
            tt.id_tweet = ttfp.id_tweet
    INNER JOIN social_network.twitter_filtro_palabras tfp ON
            ttfp.id_filtro_palabra = tfp.id
    WHERE tt.created BETWEEN
        '2017-05-20 00:00:00' AND '2017-05-22 23:59:59') tabla
GROUP BY tiempo
```

Visualización

En la gráfica de la Ilustración 73, que representa el resultado del Query 3, se puede observar el pico el día 21 de mayo del 2017, con 300 menciones ese día.

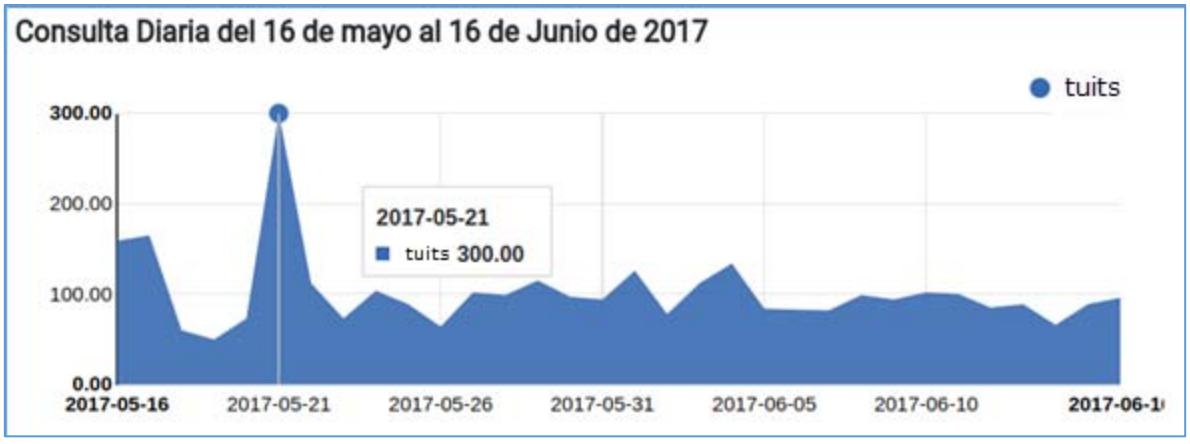


Ilustración 73. Menciones diarias del 16 de mayo al 16 de junio del 2017.

En la gráfica de la Ilustración 74, que representa el resultado del Query 4, podemos observar la distribución de las menciones en horas. El pico se encuentra entre las 23:00 horas del viernes a las 00:00 horas del sábado, con 23 menciones. Igualmente podemos apreciar que a partir del pico, 24 horas posteriores se tuvo un alza en las menciones.

Ilustración 74. Distribución de menciones por horas del 20 al 22 de Mayo del 2017.

5.5 ¿Se podrá detectar si los usuarios consumidores son adictos?

Análisis

El Query 5 realiza una consulta entre la tabla que contiene los tuits, la que contiene las palabras encontradas en los tuits y la tabla que contiene los datos del usuario. Se realiza un conteo de las menciones realizadas por los usuarios en un periodo de 31 días, para así poder observar si hay una presencia constante de menciones por los usuarios.

```
Query 5:
SELECT screen_name,description,count(*) conteo FROM (
    SELECT tu.id_user, tu.screen_name, tu.real_name, tu.description
        FROM
    social_network.twitter_tweets tt
        INNER JOIN
    social_network.twitter_tweets_filtro_palabra ttfp ON
        tt.id_tweet = ttfp.id_tweet
    INNER JOIN
        social_network.twitter_user tu ON
            tu.id_user = tt.id_user
        WHERE tt.created BETWEEN
            '2017-05-16 00:00:00' AND '2017-06-16 23:59:59') tabla
GROUP BY screen_name,description ORDER BY conteo DESC LIMIT 100
```

Visualización

Analizando los datos capturados se determina que no se pueden detectar usuarios adictos, ya que la gran mayoría de menciones son realizadas principalmente, por centros de distribución de mariguana y grupos a favor del consumo de drogas legalmente, como se puede ver en la descripción de los usuarios de twitter con mayor número de menciones.

Por otra parte, en la Ilustración 75, se puede ver el principal problema respecto a las palabras buscadas, ya que específicamente *crystal* es sinónimo de la metanfetamina, provocando esto, que el primer resultado sea una joyería, la cual entre sus tuits utiliza mucho la palabra *crystal.*

screen_name	description	conteo
Rhinestonediva	I LOVE vintage costume jewelry I collect, wear, and display my collection. I sell beautiful jewels on Etsy: Rhinestonedivas and eBay: RhinestoneDiva	190
PacificBeachMJ	Get 50% off your 1st order using code OG50. Call now 6192686658	54
MMEPHOENIX	The Medical Marijuana Exchange offers news ,products, and services for the Phoenix Medical Marijuana Community.Mobile http:t.covuFdGzd3Be18007041263	25
MMESANANTONIO	The San Antonio Medical Marijuana Exchange offers news ,products, and services for the Medical Marijuana Community 18007041263	16
intorehab	We don't know why we love it... and we don't care Snarky stories about the drugs & bad behavior of celebrities, athletes, politicians and the rest of us.	15
CannabisCard	Marijuana Medical Card is a patient assistance service. We can help you obtain a marijuana medical card	15
norrisgarman	The question is, will it blend	15
DrizlyPromo	Get $20 CREDITHave your alcohol delivered Browse the best selection of beer, wine, and liquor in one easy to shop place. Enter Promo Code at checkout	15
SDPmanagement	Worry Free #PropertyManagement #RealEstate in #SanDiego Since 2004 BBB A Accredited Free Rental Analysis Click Here: https:t.coifCz5nEqrT	14
MMESCOTTSDALE	The Scottsdale Medical Marijuana Exchange offers news	12

Ilustración 75. Usuarios y su descripción con el mayor número de menciones.

5.6 ¿Se podrán detectar distribuidores de drogas?

De acuerdo a los resultados de la pregunta anterior, no se pudieron detectar usuarios adictos, pero si se pudieron detectar centros de distribución de drogas de manera legal, principalmente de mariguana como el Medical Marijuana Exchange (MME) con sede en Phoenix con el usuario MMEPHOENIX y en San Antonio con el usuario MMESANANTONIO, que se pueden observar en la ilustración anterior.

5.7 ¿Se podrá visualizar la presencia de drogas por día en determinadas áreas geográficas?

Análisis

El Query 6 realiza una consulta entre los tuits y las apariciones de las palabras en los mismos, en un periodo de 31 días y agrupando por ciudades, para poder obtener el número de menciones a través del tiempo.

```
Query 6
SELECT ciudad, fecha, count(*) conteo FROM (
    SELECT
        ciudad, SUBSTR(created, 0, 10) AS fecha
    FROM
        social_network.twitter_tweets tt
    INNER JOIN
        social_network.twitter_tweets_filtro_palabra ttfp ON
            tt.id_tweet = ttfp.id_tweet
    WHERE tt.created BETWEEN
        '2017-05-16 00:00:00' AND '2017-06-16 23:59:59') tabla
GROUP BY ciudad, fecha
```

Visualización

Visualizando el resultado del Query 6, podemos observar que Phoenix tiene un número alto, superando a las otras ciudades regularmente, en las apariciones de las menciones relacionadas con las drogas.

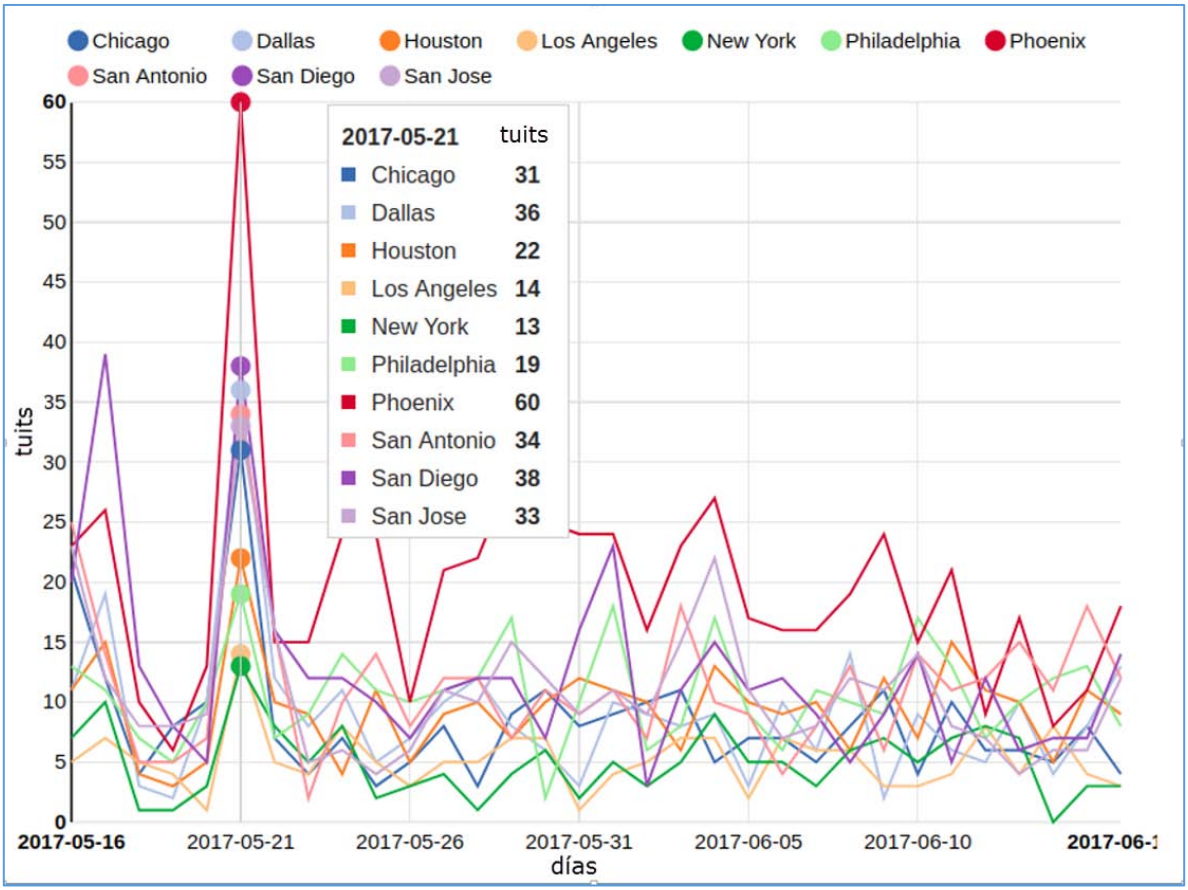


Ilustración 76. Distribución en el tiempo y las ciudades, de las menciones referentes a las drogas.

5.8 ¿Se podrá determinar en qué área está teniendo predominancia una droga en tiempo real?

Análisis

Concentrándonos en el fin de semana con el mayor número de menciones, del 19 al 21 de mayo del 2017, se obtiene la distribución de las menciones por horas entre las ciudades con el Query 7.

Query 7

```
select ciudad,tiempo, count(*) conteo from (
SELECT
    ciudad, SUBSTR(created, 0, 13) AS tiempo
FROM
    social_network.twitter_tweets tt
        INNER JOIN
    social_network.twitter_tweets_filtro_palabra ttfp ON tt.id_tweet =
ttfp.id_tweet
        where tt.created between '2017-05-19 00:00:00' and '2017-05-21
23:59:59') tabla group by ciudad, tiempo
```

Con el Query 8 se obtiene el concepto general de las palabras agrupadas por las ciudades, para poder determinar qué drogas se consumieron en el fin de semana con mayor número de menciones registradas.

Query 8

```
SELECT ciudad,palabra_concepto_general, count(*) conteo FROM (
    SELECT
        ciudad,palabra_concepto_general
    FROM
        social_network.twitter_tweets tt
    INNER JOIN
        social_network.twitter_tweets_filtro_palabra ttfp ON
            tt.id_tweet = ttfp.id_tweet
    INNER JOIN
        social_network.twitter_filtro_palabras tfp ON
            ttfp.id_filtro_palabra = tfp.id
    WHERE tt.created BETWEEN
        '2017-05-19 00:00:00' and '2017-05-21 23:59:59') tabla
GROUP BY ciudad,palabra_concepto_general
```

Visualización

Graficando el Query 7, podemos observar en la Ilustración 77, que Chicago, Phoenix y Dallas tuvieron picos en el registro de aparición de menciones.

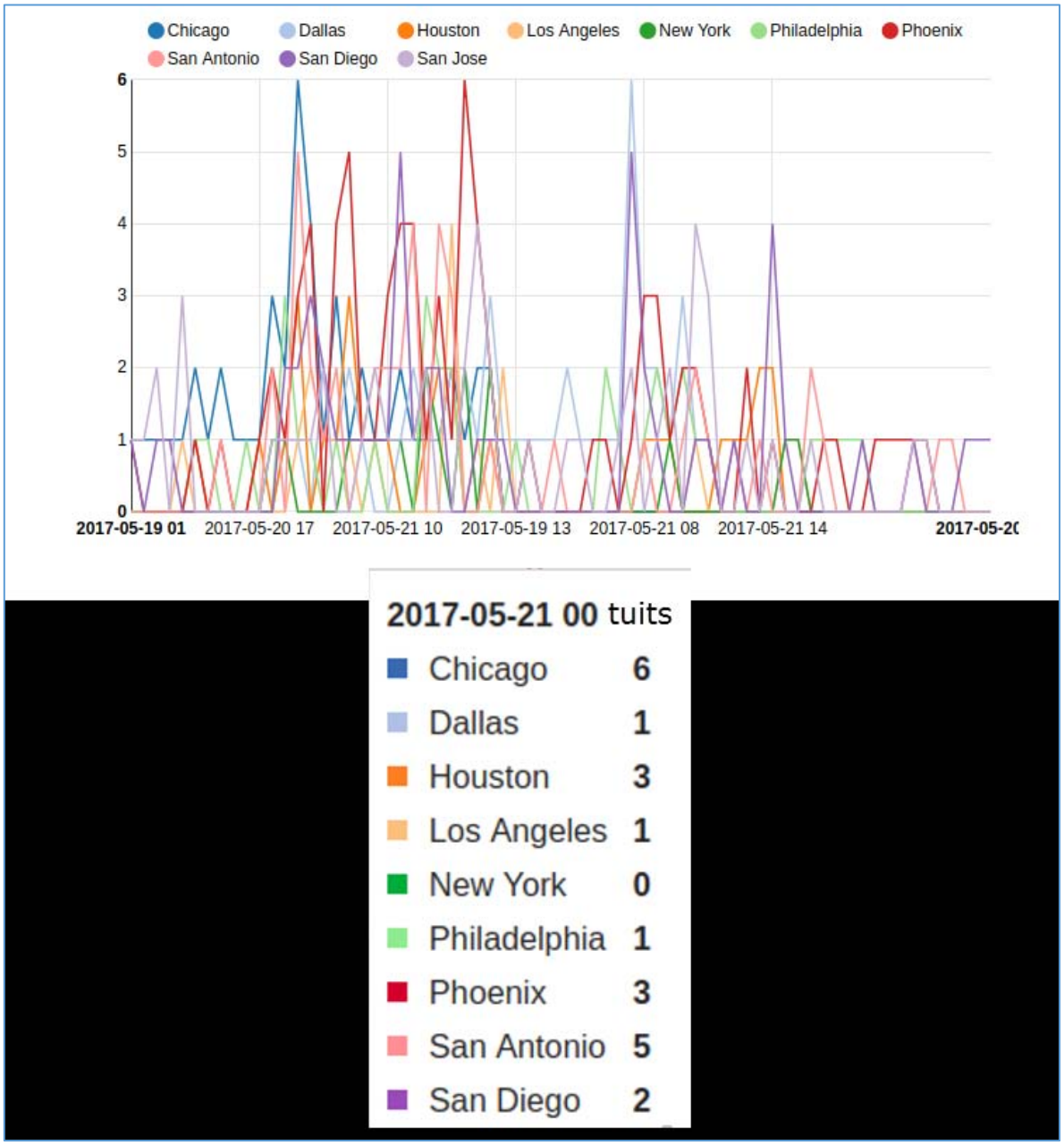


Ilustración 77. Distribución de menciones por horas entre las ciudades.

Por otro lado, en la gráfica del Query 8, podemos observar que el alcohol es lo que más aparece en las menciones, seguido de la mariguana y la cocaína. Como podemos ver en la Ilustración 78.

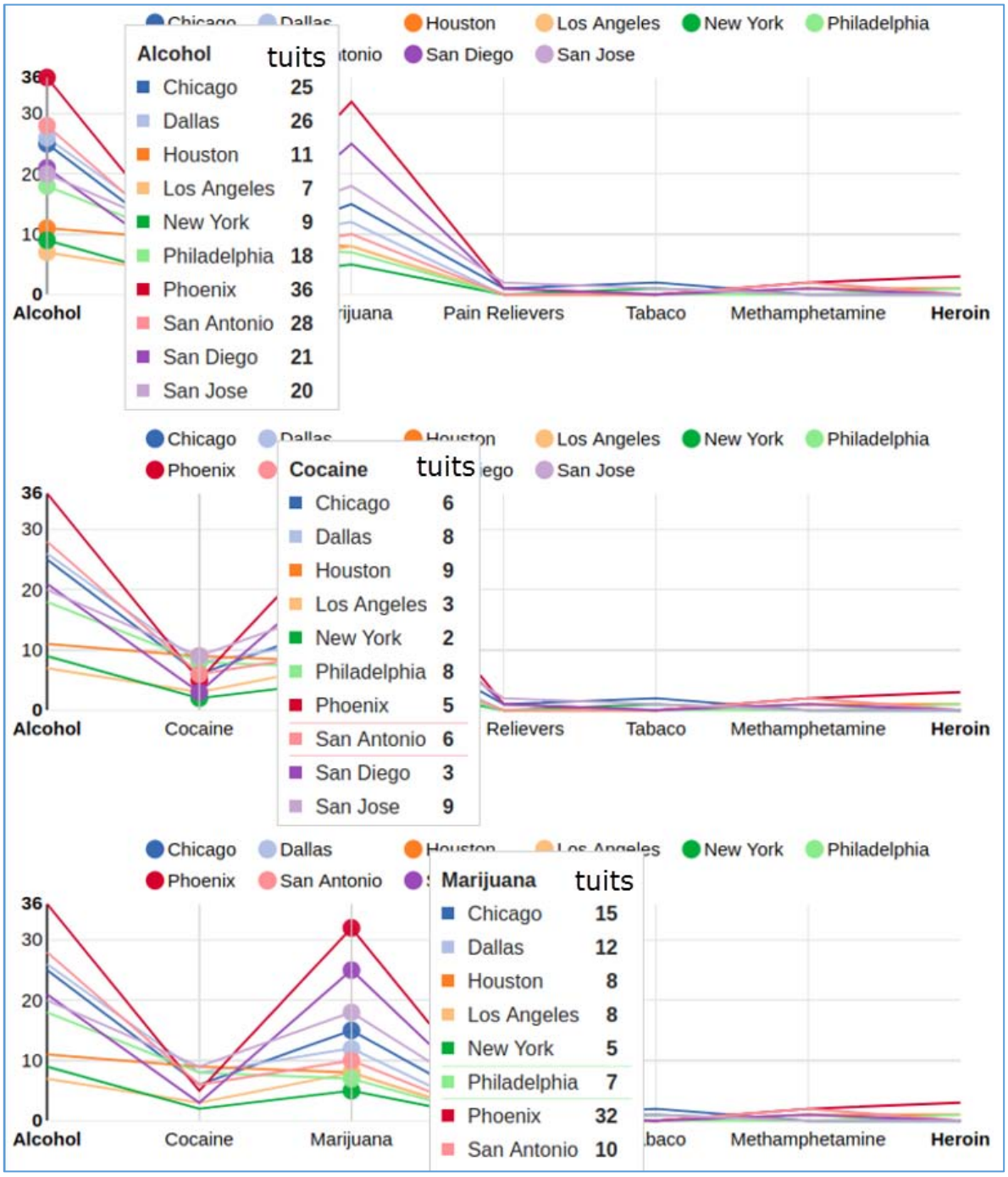


Ilustración 78. Distribución del consumo de drogas por ciudades.

5.9 En las menciones donde aparecen drogas, ¿Se podrán identificar los grupos de drogas más comunes?

Análisis

Con el Query 9, se realiza una agrupación de las palabras encontradas en los tuits, utilizando un operador de *Hive* llamado COLLECT_SET, el cual agrupa en un campo los valores únicos de una columna resultantes de una agrupación, esto para poder detectar los grupos de menciones de drogas por tuit.

Query 9

```
SELECT grupo, count(*) cuenta FROM (
    SELECT
        id_tweet,
        COLLECT_SET(palabra_concepto_general) grupo,
        COUNT(*) cuenta
    FROM
    (
        SELECT * FROM (
            SELECT
                id_tweet, palabra_concepto_general
            FROM
                social_network.twitter_tweets_filtro_palabra ttfp
            INNER JOIN social_network.twitter_filtro_palabras tfp ON
                ttfp.id_filtro_palabra = tfp.id
            GROUP BY id_tweet, palabra_concepto_general
            ORDER BY id_tweet,palabra_concepto_general
            ) tabla0
        ) tabla1
    GROUP BY id_tweet
    HAVING cuenta > 1
) tabla2
GROUP BY grupo
ORDER BY cuenta
```

Visualización

En el resultado del Query 9, que se puede observar en la gráfica de la Ilustración 79, muestra lo siguiente:

1. El alcohol y la mariguana aparecen en el primer sitio del grupo de menciones con 32 tuits.
2. La cocaína y la mariguana aparecen en el segundo puesto con 15 tuits.
3. El alcohol y la cocaína en el tercero con 14 tuits.

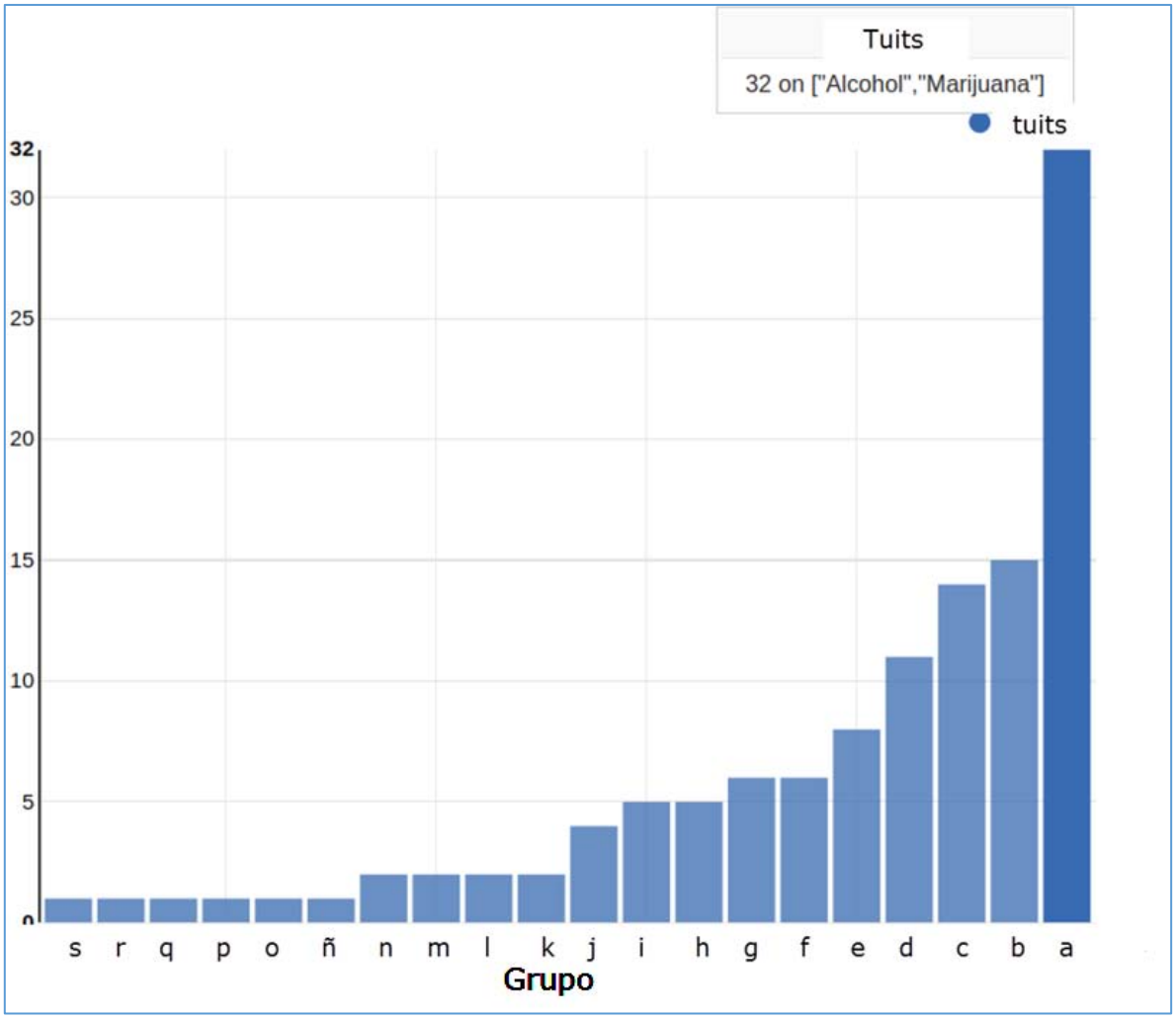


Ilustración 79. Frecuencia de los grupos de drogas que aparecen en un solo tuit.

En la Tabla 12 se puede observar el detalle de los grupos con el número de tuits por cada uno.

Tabla 12. Conteo de resultados de grupos de drogas en tuits.

Horizontal en Gráfica	Grupo	Conteo
a	["Alcohol","Marijuana"]	32
b	["Alcohol","Cocaine"]	15
c	["Cocaine","Marijuana"]	14
d	["Cocaine","Inhalants"]	11
e	["Marijuana","Pain Relievers"]	8
f	["Heroin","Marijuana"]	6
g	["Alcohol","Tabaco"]	6
h	["Cocaine","Heroin"]	5
i	["Heroin","Pain Relievers"]	5
j	["Cocaine","Methamphetamine"]	4
k	["Marijuana","Methamphetamine"]	2
l	["Alcohol","Pain Relievers"]	2
m	["Heroin","Methamphetamine"]	2
n	["Cocaine","Pain Relievers"]	2
ñ	["Marijuana","Tabaco"]	1
o	["Alcohol","Inhalants"]	1
p	["Inhalants","Marijuana"]	1
q	["Cocaine","Heroin","Marijuana"]	1
r	["Cocaine","Heroin","Pain Relievers"]	1
s	["Heroin","Inhalants"]	1

5.10 Aplicando análisis de sentimiento, ¿Se podrá obtener el resultado de la agrupación, usando solamente los tuits con menciones de drogas en general y por ciudad?

Análisis

Con el Query 10 y agrupando sobre el campo de *sentimiento*, la consulta es muy similar a las anteriores, por lo tanto solamente cambia el campo que será afectado por el operador GROUP BY y sobre la totalidad de los registros capturados.

```
Query 10
SELECT
    palabra,sentimiento, COUNT(*) cuenta
FROM
      (
      SELECT
            id_tweet,
            sentimiento,
```

```
            collect_set(palabra_concepto_general) palabra,
            COUNT(*) cuenta
    FROM
      (
      SELECT
            *
            FROM
            (
            SELECT
                  ttfp.id_tweet,
                  tfp.palabra_concepto_general,
                  tt.sentimiento
            FROM
                  social_network.twitter_tweets_filtro_palabra ttfp
            INNER JOIN social_network.twitter_filtro_palabras tfp ON
                  ttfp.id_filtro_palabra = tfp.id
            INNER JOIN social_network.twitter_tweets tt ON
                  tt.id_tweet = ttfp.id_tweet
            GROUP BY
                  ttfp.id_tweet,
                  tfp.palabra_concepto_general,
                  tt.sentimiento
            ORDER BY
                  ttfp.id_tweet,
                  tfp.palabra_concepto_general) tabla0
            ) tabla1
      GROUP BY id_tweet,sentimiento
      HAVING cuenta > 1
      ) tabla2
GROUP BY palabra,sentimiento
ORDER BY cuenta
```

El Query 11 realiza una agrupación por ciudad, las drogas en su concepto general y el sentimiento filtrando entre los 31 días de la captura de tuits, y la ciudad, para conocer a detalle las apariciones de las menciones y su sentimiento por ciudad.

```
Query 11
SELECT
      ciudad,
      palabra_concepto_general,
      sentimiento,
      COUNT(*) conteo
FROM
      social_network.twitter_tweets tt
      INNER JOIN
            social_network.twitter_tweets_filtro_palabra ttfp ON
```

```
                  tt.id_tweet = ttfp.id_tweet
      INNER JOIN
            social_network.twitter_filtro_palabras tfp ON
                  ttfp.id_filtro_palabra = tfp.id
WHERE
      tt.created BETWEEN
      '2017-05-16 00:00:00' AND '2017-06-16 23:59:59'
      AND tt.ciudad='[NOMBRE DE LA CIUDAD]'
GROUP BY ciudad , palabra_concepto_general , sentimiento
```

Visualización

Visualizando los resultados del Query 10, podemos observar en la Ilustración 80, que el grupo que contiene al alcohol y la mariguana, tienen 27 menciones con resultado negativo y 3 con resultado neutral.

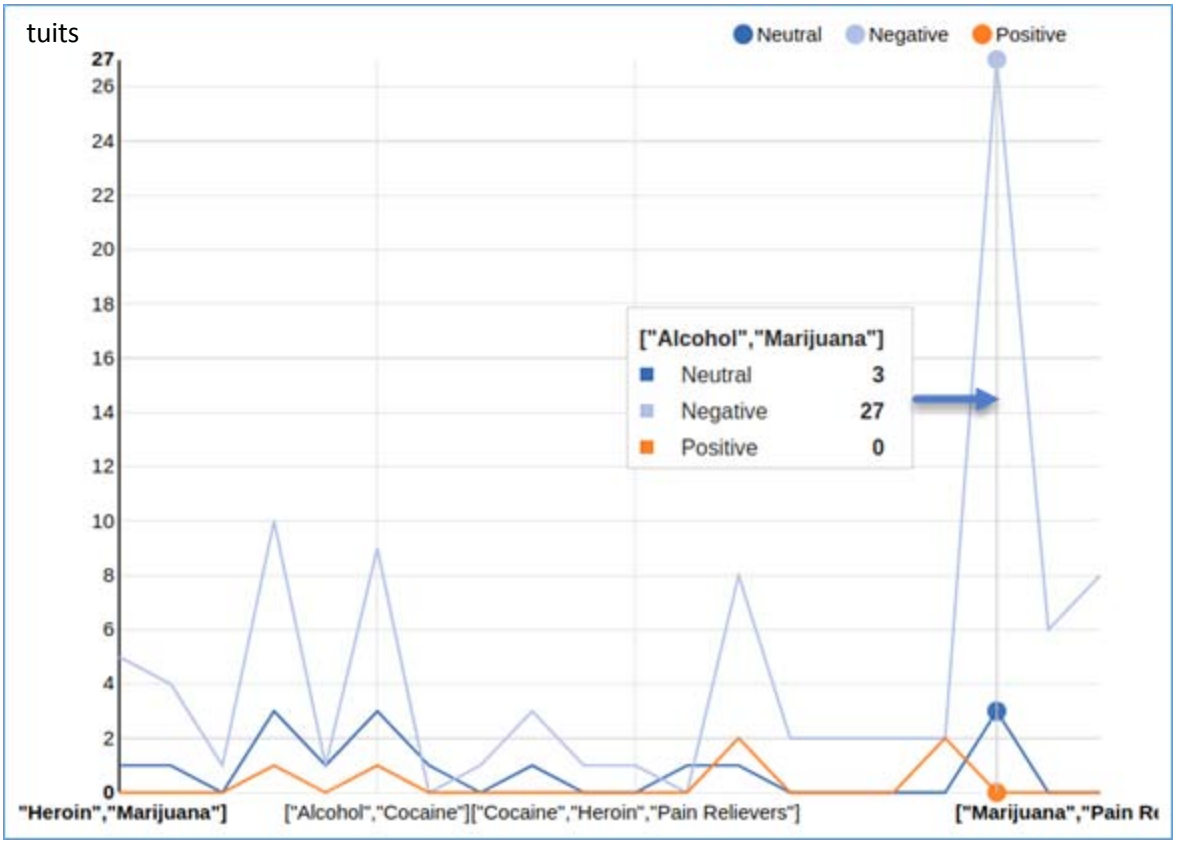


Ilustración 80. Análisis de sentimiento por grupos de drogas mostrando el detalle del alcohol y la mariguana.

En la Ilustración 81, podemos observar que la cocaína y los inhalantes tienen 8 resultados negativos, 1 neutral y 2 positivos en su sentimiento.

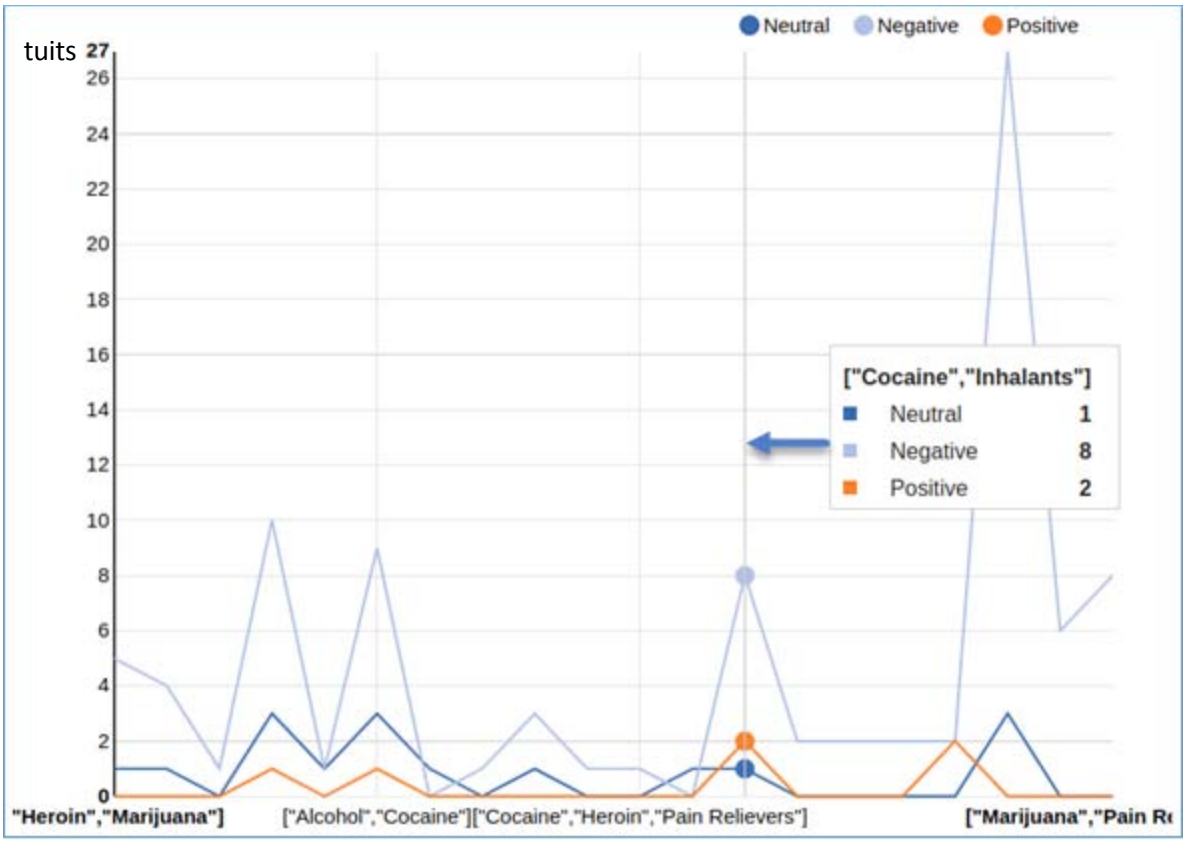


Ilustración 81. Análisis de sentimiento por grupos de drogas mostrando el detalle de la cocaína y los inhalantes.

En la Ilustración 82, podemos observar que la cocaína y la mariguana tienen 10 resultados negativos, 3 neutrales y 1 positivo.

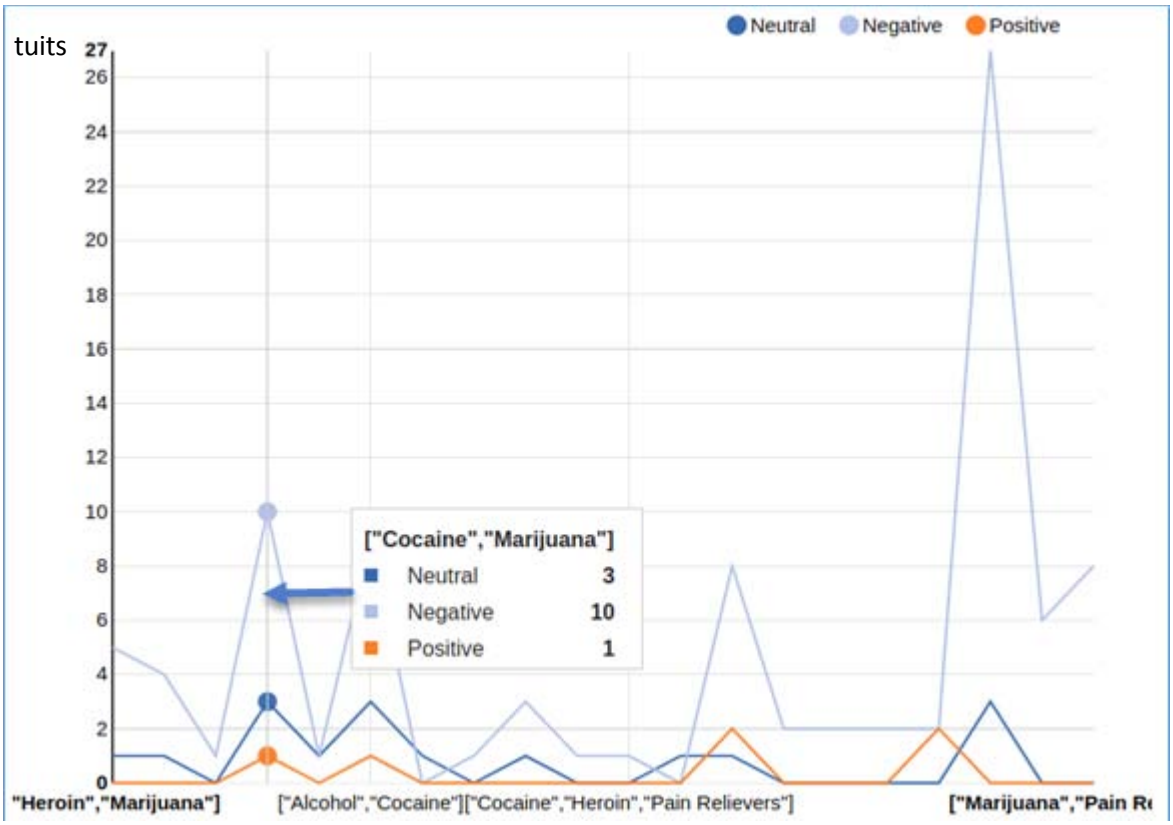


Ilustración 82. Análisis de sentimiento por grupos de drogas mostrando el detalle de la cocaína y la mariguana.

En la Ilustración 83 se puede observar el comportamiento de la aparición de menciones y su sentimiento en Chicago, Nueva York, Los Angeles y Houston.

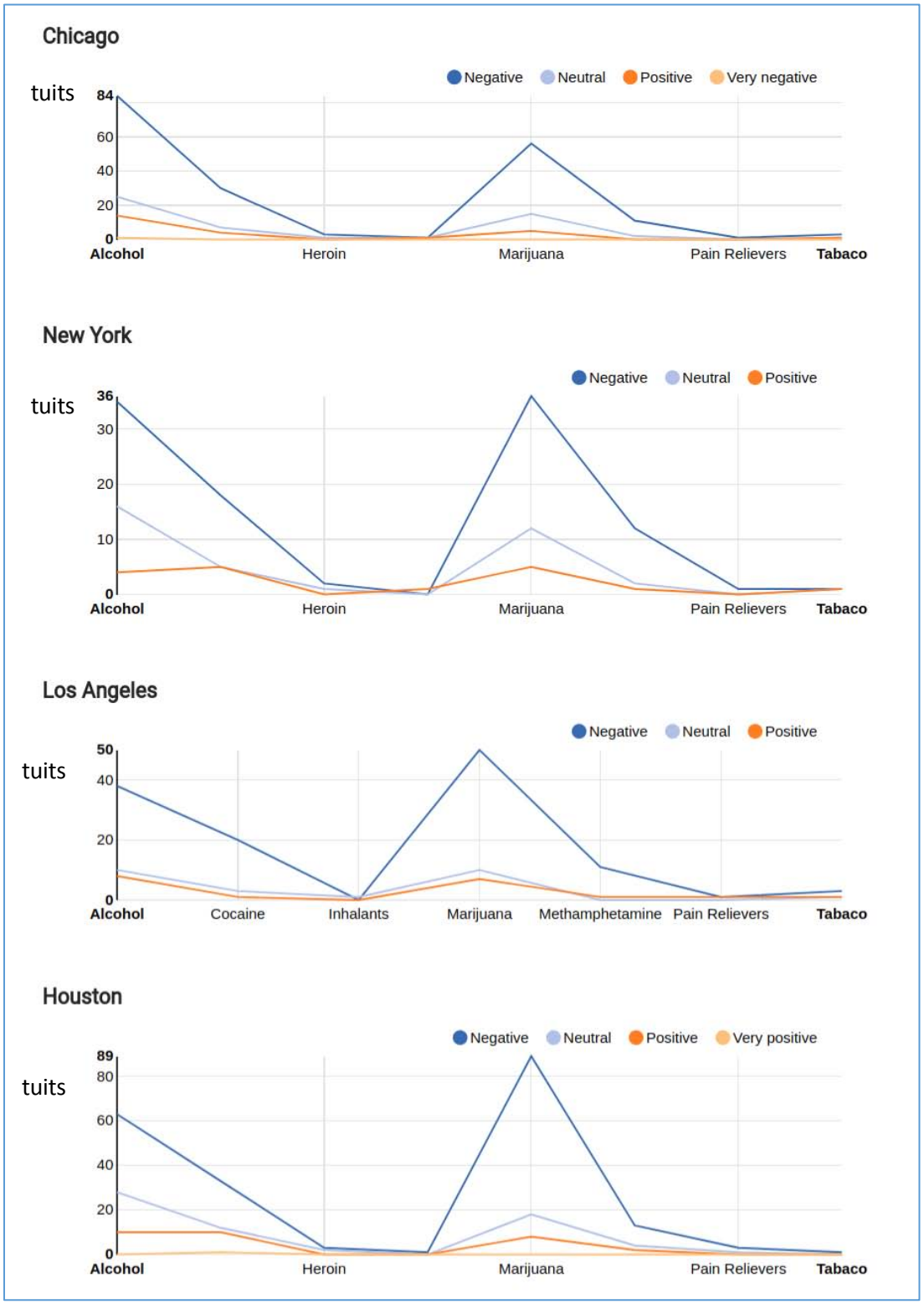


Ilustración 83. Sentimiento por droga en las ciudades de Chicago, Nueva York, Los Angeles y Houston.

En la Ilustración 84 Ilustración 83se puede observar el comportamiento de la aparición de menciones y su sentimiento por droga en Philadelphia, Phoenix, San Antonio y San Diego.

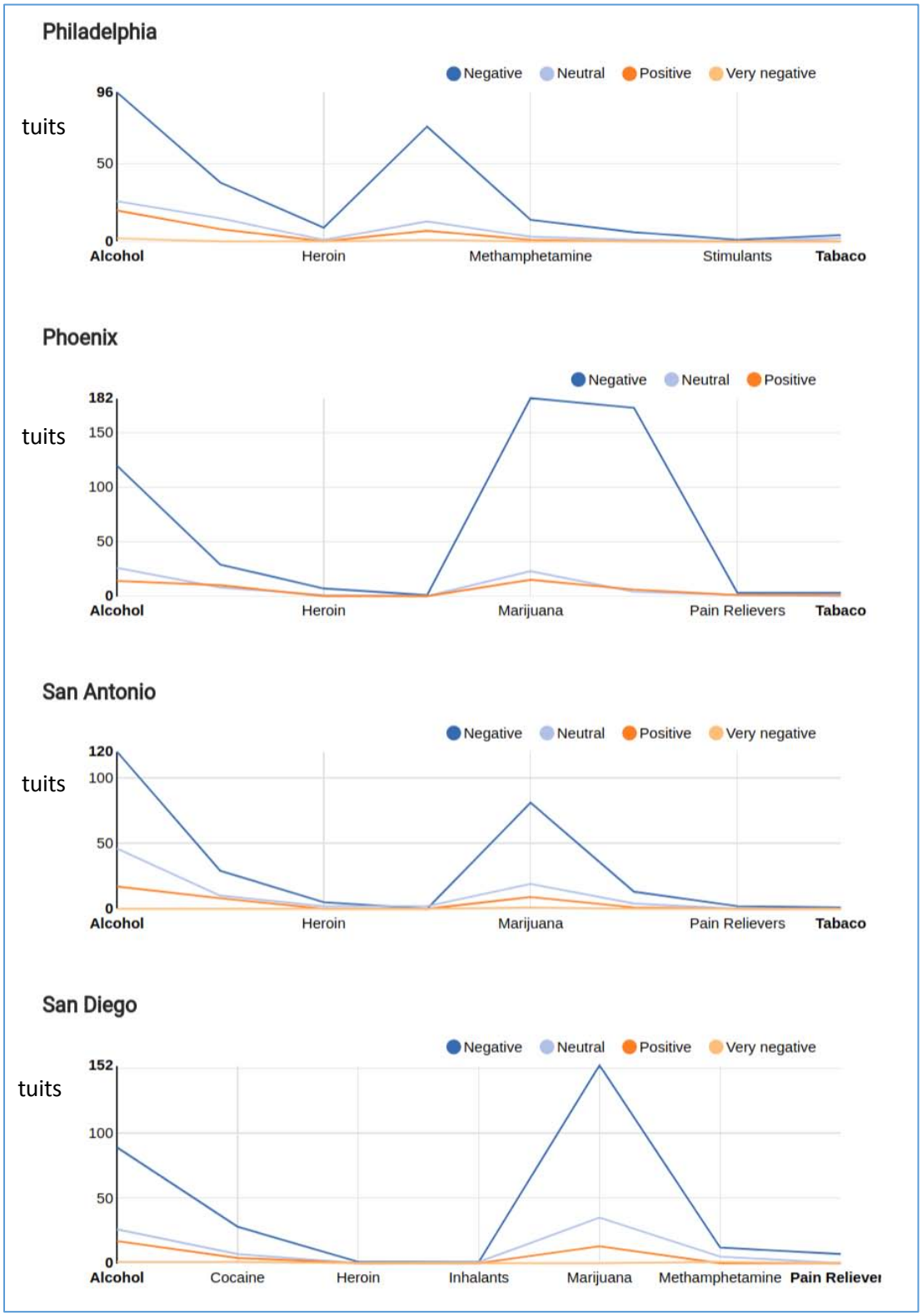


Ilustración 84. Sentimiento por droga en las ciudades de Philadelphia, Phoenix, San Antonio y San Diego.

En la Ilustración 85 se puede observar el comportamiento de la aparición de menciones y su sentimiento por droga en Dallas y San Jose.

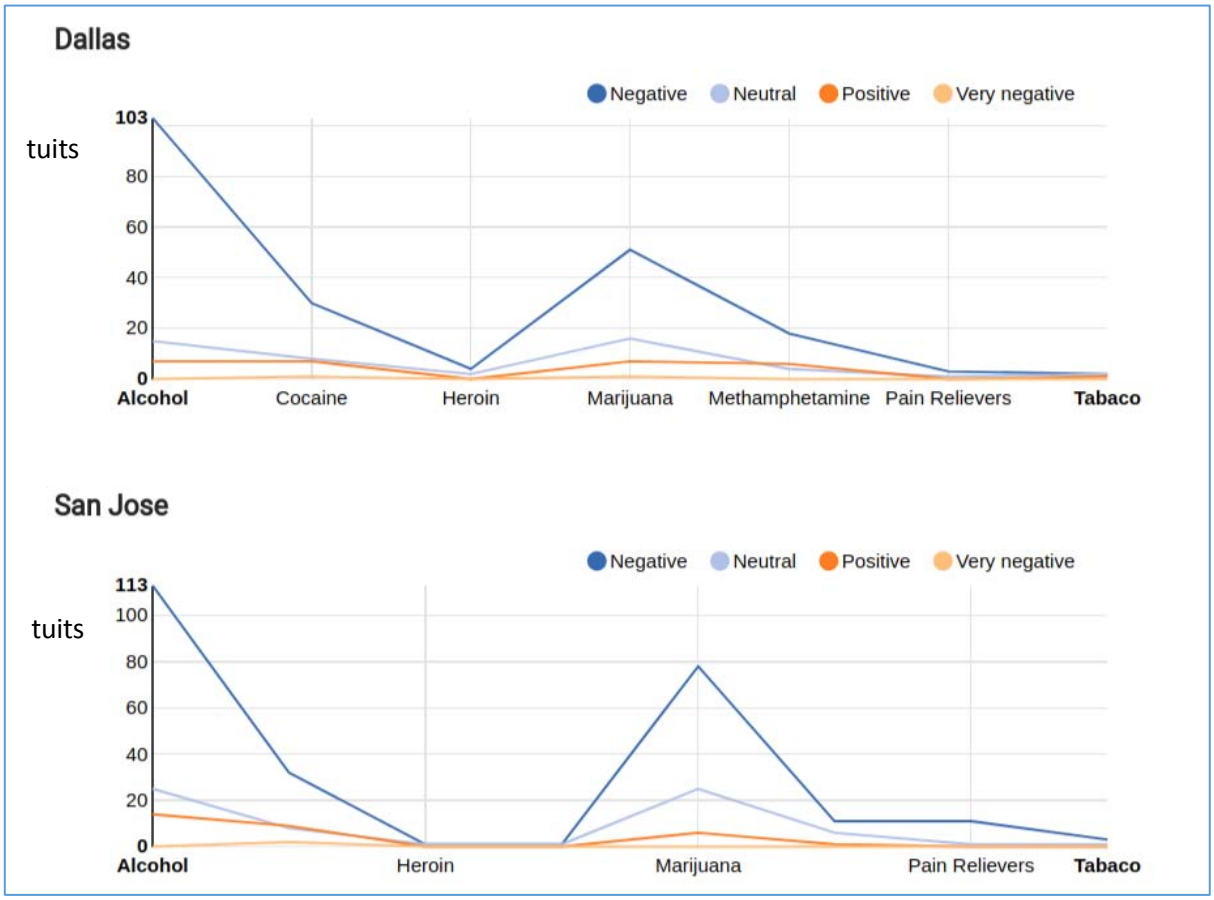


Ilustración 85. Sentimiento por droga en las ciudades de Dallas y San Jose.

5.11 De los 10 usuarios con el mayor número de menciones relacionadas con drogas, ¿Se podrá saber cuántas de sus menciones en total contienen drogas y cuantas no?

Análisis

Para obtener el resultado, el Query 12 obtiene la diferencia de menciones entre las que contienen drogas y las que no de cada usuario, primero se obtuvieron los identificadores de los 10 usuarios con mayor número de menciones referentes a drogas, para después obtener el número de tuits con menciones referentes a drogas, y la totalidad de tuits que publicaron.

```
Query 12
SELECT
    tuits,id_user, conteo
FROM
    (
    SELECT
        *
    FROM
        (
        SELECT
            'todos' tuits,
            tt.id_user,
            tu.real_name,
            tu.screen_name,
            tu.description,
            COUNT(*) conteo
        FROM
            social_network.twitter_tweets tt
        INNER JOIN social_network.twitter_user tu ON
            tu.id_user = tt.id_user
        WHERE
            tt.id_user IN (
            '796145826376450048' , '2588218549', '2675239992',
            '844840365844709378', '65244622', '740872158',
            '2935109910', '134650010', '2588353039',
            '856951294568824832')
        AND tt.created BETWEEN
            '2017-05-16 00:00:00' AND '2017-06-16 23:59:59'
        GROUP BY
            tt.id_user, tu.real_name, tu.screen_name , tu.description
        UNION
        SELECT
            tuits, id_user, '', '', '', COUNT(*)
        FROM
            (
            SELECT
```

```
                'con drogas' tuits, tt.id_user, tt.id_tweet
            FROM
                social_network.twitter_tweets tt
            INNER JOIN
              social_network.twitter_tweets_filtro_palabra ttfp ON
                tt.id_tweet = ttfp.id_tweet
            WHERE
                tt.created BETWEEN
                    '2017-05-16 00:00:00' AND '2017-06-16 23:59:59'
                AND tt.id_user IN
                  ('796145826376450048','2588218549','2675239992',
                  '844840365844709378','65244622','740872158',
                  '2935109910','134650010','2588353039',
                  '856951294568824832')
            GROUP BY tt.id_user , tt.id_tweet) tabla0
      GROUP BY id_user, tuits) tabla1
ORDER BY id_user , tuits) tabla2
```

Visualización

Al graficar los resultados, podemos apreciar la diferencia en la Ilustración 86, que hay entre la totalidad de tuit publicados por los usuarios contra los que tienen menciones referentes a drogas, en donde podemos observar que en la mayoría de los casos, en más de la mitad de sus publicaciones hacen menciones sobre drogas.

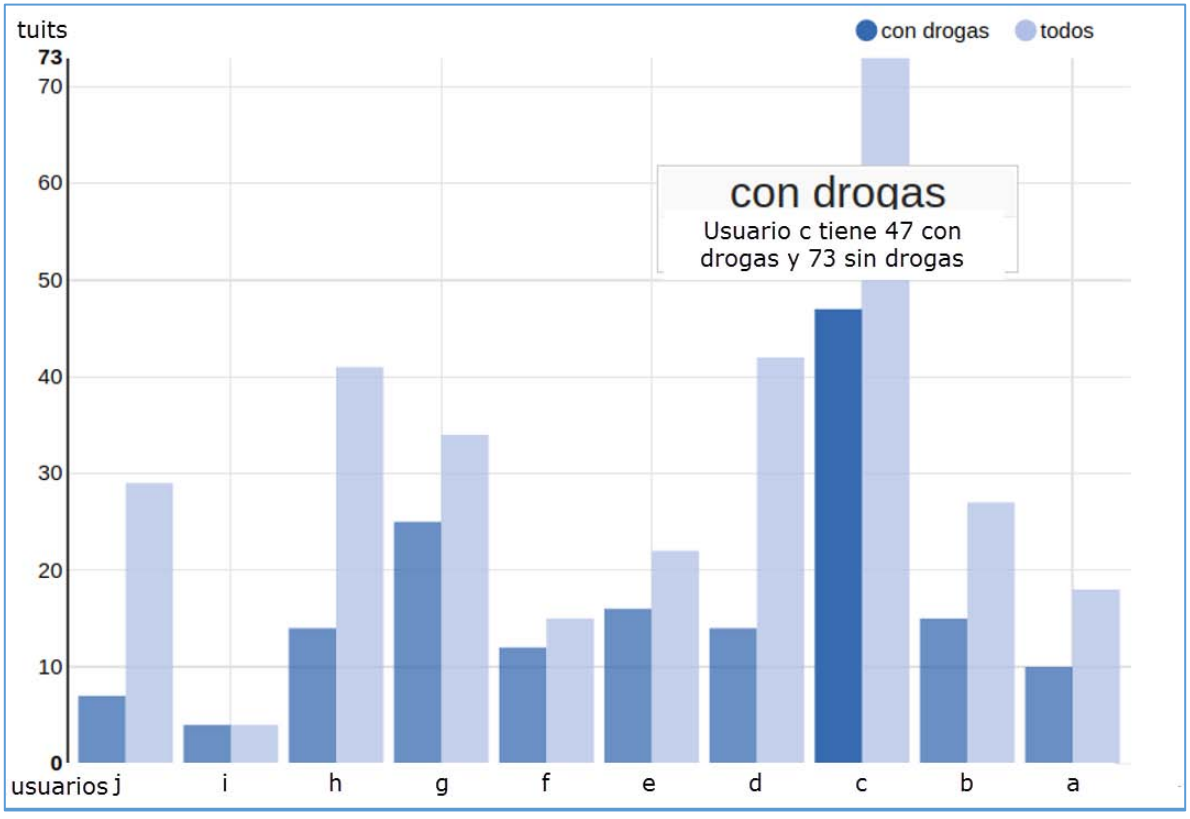


Ilustración 86. Diferencia entre la totalidad de tuits contra los que hacen mención de alguna droga.

Conclusiones

Se puede entender que existe una delgada línea entre las diferencias de un proyecto de *Big Data*, minería de datos e inteligencia de negocios, ya que su insumo en común son los datos y la diferencia es la variedad, la cantidad, los requerimientos, la velocidad y los objetivos, ya que estos elementos determinan la profundidad del análisis del proyecto que se va a implementar.

Para esta tesis, enfocada específicamente en el análisis de *Big Data*, se entiende que como base del tema, se debe identificar cuando se tienen algunas de las 3V's de velocidad, variedad y volumen. Por lo tanto, un proyecto de esta categoría, tiene un ciclo de vida similar al de las otras áreas de conocimiento relacionadas con los datos.

Realizar un proyecto de *Big Data* requiere una amplia experiencia en tecnologías, herramientas, programación, diseño e ingeniería de software, ya que una solución que se plantee, deberá de tener un diseño y una arquitectura, y sobre todo usar las herramientas adecuadas. Por poner un ejemplo, el análisis de datos masivos, sobre cantidades de datos en terabytes, en una base de datos relacional convencional, no se podría realizar en tiempo real por que puede provocar problemas en el rendimiento, ya que no está diseñada su arquitectura para ese fin. En este proyecto al ser académico, por razones de costos y proveedores de datos, no se utilizaron grandes volúmenes de datos, aunque si se utilizaron las herramientas con esa capacidad de procesamiento y se explotó en gran medida el software libre.

Por otra parte, la arquitectura de *Hadoop* que está basada en sistemas de archivos, y que al igual que un manejador de base de datos tradicional, en donde se pueden implementar consultas para el análisis de datos en paralelo, y se pueden distribuir los datos en equipos de cómputo descentralizados, en *Hadoop* cada uno de los equipos de cómputo realiza las operaciones de análisis, reduciendo así el tráfico de datos en la red, más aparte, no se utiliza para realizar transacciones de datos, esto quiere decir que una vez insertados los datos, no se pueden actualizar o eliminar registros específicos, ya que los datos son exclusivamente para realizar análisis.

El ciclo de vida de un proyecto de *Big Data*, es muy importante, ya que una mala decisión implica costos. Por ejemplo, cuando se utilizan las instancias virtuales de Amazon, y no se elige la arquitectura correcta o no se tiene la experiencia en el uso de las herramientas, se pueden elevar los costos excesivamente ya que estos servicios cobran por el tiempo que se use cada uno, desde el procesamiento, el espacio y uso de disco duro, y el tráfico de la red.

Una vez que se tienen los datos en el marco de trabajo, se procede al análisis sobre los datos contenidos. A pesar de que se puede diseñar cómo obtener los resultados, lo cierto es que saldrán nuevas necesidades o se detectarán patrones inesperados en los resultados. En este proyecto en particular, se plantean preguntas a responder como análisis inicial, pero puede ser el caso en el que no haya preguntas pero si objetivos, como por ejemplo, detectar la razón de deserción de los estudiantes o la caída en las ventas de un súper mercado, por lo que existe el caso de analizar grandes cantidades de datos para detectar patrones ocultos, es ahí cuando se puede utilizar la tecnología de *Big Data* orientada hacia la minería de datos.

La interpretación de los resultados y la visualización de los mismos es muy importante, ya que sobre el entendimiento de los resultados se tomarán las acciones para mejorar procesos, prevenir

eventos, pronosticar, etc., por lo que la tarea de elegir la mejor manera de mostrar los resultados es muy importante, ya que una mala interpretación puede llevar a sentir que el proyecto fue un fracaso o a tomar una mala decisión.

Ahora bien, hablando del tema del consumo de drogas tratado en esta tesis y tomando en cuenta que los resultados se compararon con la encuesta realizada a los habitantes de EUA, se puede observar que hay un comportamiento similar en la escala de las drogas más comunes, en otras palabras, se pudo observar en las gráficas que las tendencias son muy similares en el consumo, y sobre todo, se pudo ver en el tiempo el momento en el que se presenta el mayor índice de menciones de drogas en twitter, lo cual puede permitir a las ciudades prestar más atención y ofrecer mayor seguridad a las personas que conviven con personas que consumen drogas.

La integración con el API de análisis de sentimiento de textos es algo que incrementa el valor de los resultados, ya que se generan mayores datos y se puede tener una apreciación del sentir de un grupo de personas sobre un determinado tema. Por ejemplo, el potencial de este análisis puede predecir tendencias políticas, comerciales y sociales, ya que entender el sentir de las personas sobre un tema puede hacer entender el éxito o el fracaso de algo que se esté estudiando. Mas haya de ver comportamientos en las redes sociales, al tener entonces detectados los patrones, como trabajo futuro, se puede ampliar este proyecto y predecir el consumo de cualquier cosa, como la comida o bebidas de alguna marca, ya que comparando con estudios pasados de los cuales se defina el patrón, se puede programar entonces la búsqueda del patrón detectado para anticipar las tendencias de cualquier objeto de estudio.

Dado que en el presente trabajo de tesis se pueden obtener flujos de datos masivos de twitter, y concentrarlos en el marco de trabajo para análisis. Se tiene la capacidad para implementar proyectos de *Big Data* en cualquier contexto que se requiera, ya sea de salud, comercial, político, etc. El trabajo presentado plantea una arquitectura diseñada para cambiar de contexto y realizar un proyecto de *Big Data* de manera rápida. Puede ser de utilidad para las áreas de minería de datos o para un proyecto de ciencia de datos, ya que el marco de trabajo, permite la integración de programas, en donde se puedan ejecutar aplicaciones con algoritmos especialmente diseñados, que procesen los datos para analizar con mayor profundidad los mismos, y poder detectar patrones y conocimientos ocultos. Por lo tanto otro trabajo futuro a realizar, puede ser la aplicación de técnicas de minería de datos sobre *Big Data* en las redes sociales.

Bibliografía

A collaboration between GOOD and Oliver Munday, in collaboration with IBM. (2010). *The Daily Good*. Retrieved from The World of Data We're Creating on the Internet: https://www.good.is/infographics/the-world-of-data-we-re-creating-on-the-internet

Apache Software Foundation. (2017). *Apache Kafka*. Retrieved from https://kafka.apache.org/

Apache Software Foundation KafKa. (2016). *Apache kafka*. Retrieved from https://kafka.apache.org/intro.html

Apache Software Foundation NiFi. (2016). *Apache NiFi*. Retrieved from https://nifi.apache.org/docs.html

Apache Software Foundation Storm. (2016). *Apache Storm*. Retrieved from http://storm.apache.org/index.html

Boyd, D., & Crawford, K. (2012). CRITICAL QUESTIONS FOR BIG DATA. *Information, Communication & Society*.

Boyd, D., & Kate, C. (2011). Six Provocations for Big Data. *Oxford Internet Institute's*.

Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. *OSDI*.

Erl, T., Buhler, P., & Khattak, W. (2016). *Big Data Fundamentals: Concepts, Drivers & Techniques.* Prentice Hall.

Greenwood, S., Perrin, A., & Duggan, M. (2016). *Social Media Update 2016*. Retrieved from pewresearch.org: http://assets.pewresearch.org/wp-content/uploads/sites/14/2016/11/10132827/PI_2016.11.11_Social-Media-Update_FINAL.pdf

Hanson, C. L., Burton, S. H., Giraud-Carrier, C., West, J. H., Barnes, M. D., & Hansen, B. (2013). Tweaking and Tweeting: Exploring Twitter for Nonmedical Use of a Psychostimulant Drug (Adderall) Among College Students. *JOURNAL OF MEDICAL INTERNET RESEARCH*.

Hortonworks. (2016). *Hortonworks.com*. Retrieved from http://hortonworks.com

Konstantin, S., Hairong, K., Sanjay, R., & Robert, C. (2010). The Hadoop Distributed File System. *IEEE*.

Laney, D. (2001). 3D Data Management: Controlling Data Volume, Velocity, and Variety. *META Group Inc.*

Li, J., Fei, T., Ying, C., & Liangjin, Z. (2014). Big Data in product lifecycle management. *Springer*.

Martínez Cámara, E., Martín Valdivia, M., & Ureña, L. (2011). *Análisis de Sentimientos.* Retrieved from Red Temática en Tratamiento de la Información Multilingüe y Multimodal (TIMM): http://timm.ujaen.es/wp-content/uploads/2014/03/analisis_de_sentimientos.pdf

McClellan, C., Ali, M. M., Mutter, R., Kroutil, L., & Landwehr, J. (2016). Using social media to monitor mental health discussions - evidence from Twitter. *Journal of the American Informatics Association*.

Min Chen, S. M. (2014). Big Data: A Survey. *Springer Science+Business Media New York 2014*. Retrieved from ECC: http://www2.egr.uh.edu/~zhan2/ECE6111/class/BigDataSurvey2014.pdf

Moore, G. (1964). The Future of Integrated Electronics. *Fairchild Semiconductor internal publication*.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). *SemEval-2016 Task 4: Sentiment Analysis in Twitter.* Retrieved from International Workshop on Semantic Evaluation 2016: http://alt.qcri.org/semeval2016/task4/

Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008). Pig Latin: A Not-So-Foreign Language for Data Processing. *ACM*.

Socher, R., Perelygin, A., Y. Wu, J., Chuang, J., D. Manning, C., Y. Ng, A., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).*

Substance Abuse and Mental Health Services Administration. (2015). *Results from the 2015 National Survey on Drug Use and Health.*

Thusoo, A., Joydeep, S., Namit, J., Zheng, S., Prasad, C., Ning, Z., . . . Raghotham, M. (2010). Hive – A Petabyte Scale Data Warehouse Using Hadoop. *ICDE Conference 2010*.

Tutorialspoint. (2017). *Tutorialspoint*. Retrieved from https://www.tutorialspoint.com/map_reduce/

U.S. Census Bureau. (2010). *Census*. Retrieved from https://www.census.gov/2010census/

WEForum. (2012). *The World Economic Forum.* Retrieved from https://www.weforum.org/reports/big-data-big-impact-new-possibilities-international-development/

West, J., Hall, P., Prier, K., Hanson, C. L., Giraud-Carrier, C., Neeley, E., & Barnes, M. D. (2012). Temporal variability of problem drinking on Twitter. *Open Journal of Preventive Medicine*.

Zimbra, D., Ghiassi, M., & Lee, S. (2016). Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks. *IEEE Computer Society*.