Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Una plataforma base para *Big Data*

Tesis que presenta

José Juan Martínez Peláez

para obtener el Grado de

Maestro en Ciencias en Computación

Director de Tesis

Dr. Jorge Buenabad Chávez

México, D.F. **Diciembre de 2015**

Resumen

Big Data es el término usado desde la decada pasada para referirse al análisis de datos en grandes cantidades, de diferentes tipos, o ambos, con el propósito de ayudar a la toma de decisiones. También se refiere a las herramientas de *software* para realizar tal análisis, particularmente *MapReduce*, el modelo de programación y ambiente de ejecución desarrollado por *Google* para procesar grandes cantidades de datos en paralelo y así reducir el tiempo de respuesta. Aunque la mayoría de tales herramientas son libres y abiertas, su complejidad es tal que no es trivial instalarlas ni utilizarlas en conjunto. Por esta razón un proyecto *Big Data* requiere de un grupo interdisciplinario de personas: analistas, expertos del área y especialistas de software.

Esta tesis presenta *BDSP* (del inglés *Big Data Start Platform*), un sistema web en el que usuarios pueden realizar tareas de manejo y análisis de datos tipo *Big Data* desde cualquier lugar, a cualquier hora y con cualquier dispositivo con acceso a Internet y un *browser*. *Databricks* es el único sistema web similar a *BDSP*, pero es comercial. *BDSP* consiste de una interfaz de gráfica con la que usuarios especifican dichas tareas, y de los módulos que las realizan sobre un cluster de procesamiento paralelo con Hadoop, la versión libre y abierta de *Mapreduce*. *BDSP* también integra diferentes fuentes de datos externas (*Twitter*, *Facebook*, entre otras) por medio de servicios *Web*. El propósito de *BDSP* es servir como prototipo inicial de proyectos *Big Data*, como plataforma base para extenderla según se requiera, y como vehículo de capacitación en análisis de datos y en desarrollo de *software Big Data*.

BDSP es un sistema desarrollado sobre el servidor *Web* Apache *HTTP*. La interfaz de usuario es adaptable a las capacidades de visualización de cualquier dispositivo con acceso a Internet. Actualmente *BDSP* integra los paquetes *Hadoop*, *Mahout* y *NLTK*, con los que soporta los siguientes tipos de anásis: regresión, clasificación, agrupamiento y análisis de sentimiento. La tesis muestra el uso de *BDSP* en la solución de dos tipos distintos de análisis. El diseño modular de *BDSP* basado en web services permite que sus módulos puedan ser usados por aplicaciones externas y puedan ser sustituídos por otros módulos de funcionalidad equivalente.

Agradecimientos

A mis padres, Susana Peláez de la Rosa y Juan José Martínez Sánchez, por apoyarme siempre.

A mi asesor, el Dr. Jorge Buenabad Chávez por sus enseñas y observaciones.

Al MC José Rangel García por la idea del tema para realizar esta tesis.

A mis revisores de tesis, la Dra. Sonia Mendoza Chapa y el Dr. José Guadalupe Rodríguez García por tomarse el tiempo de revisar mi tesis y sus valiosos comentarios.

Al CONACyT (Consejo Nacional de Ciencia y Tecnología) por el apoyo económico brindado durante la maestría.

Al CINVESTAV-IPN (Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional) por permitirme ser parte de este prestigioso centro de investigación.

Índice general

Índice de figuras

Capítulo 1

Introducción

Big Data está siendo adoptado como una herramienta esencial para el análisis de datos en diversas áreas de las ciencias, las ingenierías, los negocios y otras áreas en donde la toma de decisiones informadas, predicción e inferencia basada en el análisis de datos hace una marcada diferencia [1]. Algunos gobiernos están desarrollando, o bien planean desarrollar proyectos de *Big Data* a gran escala [2]. No obstante, adoptar las herramientas de *Big Data* es un reto, ya que existen muchos factores involucrados en su implementación, entre otros: muchos tipos de análsis de datos, multiples tipos de datos y en grandes cantidades, multiples fuentes de datos (*Twitter*, *Facebook*, entre otros), variadas herramientas de software, y la complejidad de integrar todo ello en una plataforma fácil de usar.

El análisis de datos en el contexto de *Big Data* puede ser definido como el proceso de explorar y analizar con velocidad grandes volúmenes de datos, posiblemente de distintos tipos, con el fin de extraer información útil en forma de patrones significativos. Este análisis usualmente es llamado *BDA* (del inglés *Big Data Analytics*), también es conocido como minería de datos [3] o estadística a escala y velocidad [4]. *BDA* incluye varios tipos de análisis tales como regresión (lineal, no líneal, simple, múltiple, *etc.*), clasificación (árboles de decisión, *ID3*, *etc.*), agrupamiento (*K-Means*, F*uzze C-Means*, *etc.*), muestreo (Gibbs, Metropolis-Hasting, Monte Carlo, *etc.*) o análisis de sentimiento (bolsa de palabras, *etc.*), sólo por mencionar algunos.

BDA también incluye Procesamiento Analítico en Línea, u *OLAP* (del inglés *On-Line Analytical Processing*), y Recuperación de Información, o *IR* (del inglés *Information Retrieval*) [3]. *OLAP* ha sido ampliamente utilizado por empresas para tomar mejores decisiones de negocios, por ejemplo: qué productos almacenar en inventario y cómo publicitar para aumentar las ventas. Se basa en consultas *SQL* a datos estructurados en una base de datos tipo *datawarehouse*, un repositorio central con información de múltiples fuentes, en el que se incluyen datos transaccionales. *IR* se basa en consultas a datos no estructurados, como los de páginas *Web* y otros documentos (como imágenes, audio, etc.). *IR* involucra análisis, clasificación y creación de índices de documentos basados en palabras clave con el propósito de encontrar documentos o información en los mismos que sea de interés para el usuario [5].

El término *Big Data* posiblemente se comenzó a utilizar justo después de la publicación de *MapReduce* [6], el modelo de programación y plataforma de ejecución diseñados y utilizados por *Google* "para la generación de datos para el servicio web de búsqueda de *Google*, y muchos otros sistemas" [6]. *Google* también utiliza *MapReduce* para realizar ordenamientos, análisis de minería de datos, procesos de aprendizaje de máquina y muchos otros servicios ofrecidos por la compañía [6, p. 12]. El término *Big Data* se pudo haber acuñado por el hecho de que el servicio de búsqueda de *Google* conlleva el procesamiento de toda la (*big*) *Web*: todos los datos disponibles públicamente en la *Web* [7].

MapReduce es una plataforma diseñada para ejecutarse en un *cluster* de computadoras de propósito general, por lo cual es económico y asequible a pequeñas y medianas empresas e instituciones. Además, *MapReduce* proporciona, de manera totalmente transparente para el programador, tolerancia a fallas y balanceo de carga haciendo posible procesar grandes cantidades de datos de manera eficiente. Es decir, sin la tolerancia a fallas incluida en la plataforma un error en el procesamiento de los datos requeriría recalendarizar el procesamiento y empezar desde cero. El modelo de programación de *MapReduce* no requiere que el programador desarrolle programas paralelos complejos. El estilo de programación es similar a interconectar aplicaciones

secuenciales por medio de *pipes* en *Unix*, pero *MapReduce* ejecuta tales aplicaciones y *pipes* en paralelo de forma totalmente transparente al programador. No obstante, los programas realizados en el modelo de programación de *MapReduce* aún son de muy bajo nivel para programar las tareas complejas requeridas por *BDA*.

Hadoop es una versión libre y abierta de *MapReduce* que fue desarrollada por *Yahoo!* — actualmente la Fundación Apache continua con su desarrollo. La creación de *Hadoop*, y posteriormente el desarrollo de varias herramientas de *software* de análsis de datos que se ejecutan sobre *Hadoop*, causaron un gran impacto en la industria de las tecnologías de la información (TI). Un ejemplo de estas herramientas es *Mahout*. Estas herramientas hicieron posible procesar volúmenes muy grandes de datos estructurados y no estructurados a un costo razonable; mientras que las soluciones basadas en la nube hicieron posible realizar el procesamiento de datos sin la necesidad de invertir grandes cantidades de dinero en la compra del *hardware* necesario.

Antes de la creación de *MapReduce*/*Hadoop*/*etc.*, sólo las grandes empresas podían permitirse hacer *BDA* usando costosos sistemas de base de datos relacionales tales como *Oralce*, *IBM DB2*, *Teradata*, entre otros, los cuales se ejecutan en sistemas de cómputo paralelo grandes que **no** eran de propósito general, y por lo tanto eran muy costosos [3]. Los sistemas *Data Appliances* como *DATAllegro* [8], *Netezza* [9], *Greenplum* [10], entre otros, hicieron que *BDA* fuera más accesible a más empresas, pero seguían siendo relativamente caros a pequeñas empresas.

Las tecnologías relacionadas con *Big Data* han sido rápidamente adoptadas por áreas como Bioinformática, Salud, Física, Economía, *eLearning*, entre otras. No obstante, un proyecto *Big Data* es complejo por todos los factores que involucra. Y será exitoso sólo si integra a personas con conocimiento del área del problema o aplicación a resolver, a personas con experiencia en el uso de análisis de datos relevantes, y a personas con experiencia en el uso de las herramientas de *software Big Data*. Además, debe considerarse también la integración de las herramientas *Big Data* con la infraestructura de tecnologías de la información existente.

Esta tesis presenta *BDSP* (del inglés *Big Data Start Platform*), un sistema *Web*

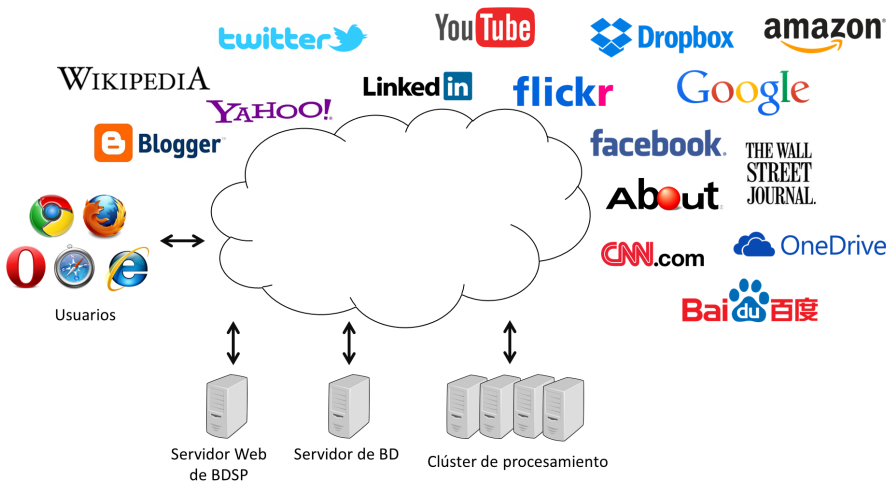

Figura 1.1: Ambiente de *BDSP*

en el que usuarios pueden realizar tareas de manejo y de análisis de datos tipo *Big Data* desde cualquier lugar, a cualquier hora y con cualquier dispositivo con acceso a Internet y un *browser*. *BDSP* integra diferentes fuentes de datos y tareas de procesamiento mediante el uso de servicios *Web*. El propósito principal de su diseño es facilitar el desarrollo de proyectos *Big Data* sirviendo de tres maneras: 1) como una herramienta para la creación rápida de prototipos; 2) como una plataforma que puede ser modificada y extendida según se necesite; y 3) como un vehículo de capacitación en análisis de datos y en desarrollo de *software Big Data*.

La figura 1.1 muestra el ambiente de *BDSP*. Los usuarios acceden a *BDSP* mediante un *browser*, a la izquierda de la figura. Los componentes funcionales de *BDSP* son mostrados en la parte inferior de la figura e incluyen: i) un servidor *Web*, cuya interfaz gráfica permite al usuario especificar y configurar tareas para manejar y analizar datos; ii) un servidor de base de datos, donde se almacena información acerca de los usuarios y otros recursos; y iii) un *cluster* de procesamiento basado en *Hadoop*. Los datos de usuarios son procesados y almacenados utilizando *HDFS* (del inglés *Hadoop Distributed File System*), el sistema de archivos distribuido y paralelo con tolerancia a fallas y balance de carga utilizado por *Hadoop*.

Los tres componentes de *BDSP* pueden ser ejecutados en la misma computado-

Figura 1.2: Descripción de las capas de *BDSP*

ra o en computadoras distintas según se necesite. Estos componentes podrían ser remplazados por otros con funcionalidad equivalente. Incluso sería posible realizar el procesamiento de los datos utilizando un *cluster* privado administrado por el usuario y que esté basado en *Hadoop* o procesarlos mediante el uso del servicio denominado *Elastic MapReduce* (*EMR*) ofrecido por la empresa *Amazon*.

Además de la información para acceder a *BDSP*, el servidor de bases de datos maneja, para cada usuario, otros nombres de usuario y contraseñas para que *BDSP* pueda acceder archivos en otras cuentas del usuario en otros sistemas accesibles a través de Internet. Los usuarios pueden subir, a *BDSP*, archivos de datos en sus cuentas de *Dropbox*, *Google Drive*, *Twitter* y *Facebook*.

Para cada usuario, conceptualmente *BDSP* puede ser visto como compuesto por las tres capas lógicas mostradas en la figura 1.2: una interfaz gráfica, o *GUI* (del inglés *graphical user interface*), una capa de análisis de datos, y una capa de manejo de datos.

La *GUI* es la interfaz del sitio *Web* y fue desarrollado en *PHP*, *MySQL*, *HTML5* y *JavaScript*. Este sitio cuenta con un Diseño *Web* Adaptable que permite una visualización óptima sin importar el dispositivo que se utilice. La biblioteca que permite esta adaptabilidad se llama *Bootstrap*.

La capa de análisis de datos consiste de servicios *Web* construidos con *PHP* cuya función es encapsular herramientas para el análisis de datos. Los usuarios pueden proporcionar parámetros y visualizar los resultados a través de la *GUI*. Los análisis de datos que *BDSP* puede ejecutar sobre *Hadoop* utilizan los *frameworks NLTK* [11] (*Natural Language Toolkit*) y *Mahout* [12].

La capa de manejo de datos consiste en servicios *Web* construidos con *PHP* y *MySQL* y su función es obtener y manejar archivos locales y remotos para almacenarlos en el *HDFS* de *BDSP*. *BDSP* puede obtener archivos remotos del usuario, almacenados en sistemas *Web* tales como *Dropbox* y *Google Drive*, e información publicada por terceros en *Twitter* y *Facebook*. Como ya se mencionó, el almacenamiento de archivos es realizado en *HFDS*.

En la figura 1.2, las flechas negras sobre las capas de análisis de datos y manejo de datos, indican que los servicios *Web* de tales capas pueden ser invocados directamente por aplicaciones externas, además de que los usuarios interactúan con las mismas mediante el uso de la *GUI*.

Actualmente *BDSP* integra los paquetes *Hadoop* [13], *Mahout* [12] y *NLTK* [11], y con los mismos, soporta los siguientes tipos de anásis: regresión, clasificación, agrupamiento y análisis de sentimiento. La tesis muestra el uso de *BDSP* en la solución de tres tipos distintos de análisis.

La versión actual de *BDSP* soporta un total de 12 análisis y una tarea con los *frameworks* que integra: *NLTK* (1 tarea) y *Mahout* (12 análisis). Sin embargo, esto se debe a que la *GUI* de *BDSP* sólo maneja actualmente la configuración de esos 12 análisis y una tarea. Claramente, la *GUI* de *BDSP* puede extenderse bastante con otros análisis de esas u otras herramientas. También se puede enriquecer aún más a *BDSP* si se agregan las funciones de visualización como las que maneja el sistema *Zepellin* [14]: gráficas de barras, circulares, de líneas, de área, entre otras.

Es importante mencionar que *BDSP* es, hasta donde sabemos, el primer sistema *Web* para *Big Data* que es libre y abierto. *Databricks* [15] es un sistema comercial en la *Web* similar a *BDSP*, que permite al usuario manejar datos y analizarlos en

un *cluster* basado en *Spark*, una versión de *MapReduce* que utiliza principalmente operaciones realizadas en memoria y que es hasta cien veces más rápido que *Hadoop*.

También, existen obviamente muchos productos para *Big Data* que no son *Web* pero que han tenido mucha aceptación y han sido usados para el análisis de datos aún antes de la llegada del término *Big Data*, tales como *RapidMiner* [16], *Pentaho* [17], *KNIME* [18] y *SPSS* [19].

1.1. Organización de la tesis

El capítulo 2 presenta una introducción al concepto de *Big Data*, muestra una breve descripción de los tipos de análisis que actualmente se realizan y algunas de las herramientas y plataformas más utilizadas.

El capítulo 3 presenta tres ejemplos de problemas que se han resuelto exitosamente utilizando técnicas de *Big Data*. Se abordará el uso que *Big Data* ha tenido en los negocios para el reconocimiento de patrones, en la bioinformática para el procesamiento de volúmenes muy grandes de información para el procesamiento del *DNA* y en el área geoespacial en donde los datos son generados de manera muy rápida por los satelites y su procesamiento es un reto.

El capítulo 4 presenta el diseño de *BDSP*. En este capítulo se presenta su organización, una descripción detallada de su arquitectura de *software* basada en tres capas y se mencionan los aspectos de seguridad que se han integrado.

El capítulo 5 presenta a *BDSP* desde la perspectiva del usuario: cómo accederlo y configurarlo y cómo usarlo en varias aplicaciones. Se resuelve un ejemplo de un análisis de agrupamiento y un ejemplo de análisis de sentimiento de datos obtenidos de *Twitter*.

El capítulo 6 presenta nuestras conclusiones y algunas ideas para trabajo futuro.

Capítulo 2

Big Data

2.1. Antecedentes

En los últimos años la cantidad de datos digitales que se almacenan diariamente se ha incrementado de manera considerable. Estudios realizados por la Corporación de Datos Internacional o *IDC* (del inglés *International Data Corporation*) estimaron que en 2007 la cantidad de datos almacenados era aproximadamente 281 exabytes. Para 2011 el volumen de los datos almacenados alcanzaba la cantidad de 1.8 zettabytes [20]. *IDC* calculó que para 2020 se habrán almacenado más de 44 zettabytes [21][1].

2.1.1. Motivación

Desde la decada pasada se ha estado utilizando el término *Big Data* para referirse al procesamiento y análisis de datos en grandes cantidades, de diferentes tipos, o ambos, y en un tiempo razonable. Laney [22] y otros se refieren al análisis de *Big Data* como caracterizado por volumen, velocidad y variedad. Pero *Big Data* también se refiere a las herramientas de *software* que se utilizan para procesar los datos, particularmente *MapReduce*, el modelo de programación y ambiente de ejecución diseñado y desarrollado por *Google* para procesar en paralelo grandes cantidades de datos en

[1] 1 zettabyte = 1024 exabytes; 1 exabyte = 1024 petabytes; 1 petabyte = 1024 terabytes; 1 terabyte = 1024 GB.

clusters configurados con *hardware* de propósito general. Es decir, a un costo muy razonable; y en la nube, sin necesidad de una inversión fuerte inicial.

Existe mucho interés en *Big Data* en los negocios, las ciencias, las ingenierías y en varias áreas más, para obtener información útil en la toma de desiciones a partir del análisis de la gran cantidad de datos que han almacenado. Muchos gobiernos en todo el mundo se encuentran también interesados en *Big Data*. Por ejemplo, en 2012 el Gobierno de Estados Unidos anunció que invertiría 200 millones de dólares en el Plan de Invesitigación y Desarrollo de *Big Data* (en inglés *Big Data Research and Development Plan*) [23]. Con este plan se pretende generar tecnología que facilite la investigaión en áreas como educación, cuidado del medio ambiente y la salud. Entre los proyectos que se contemplan en este plan se encuentra el proyecto *BD2K* (del inglés *Big Data to Knowledge*), promovido por el Instituto Nacional de Salud, o *NIH* (del inglés *National Institutes of Health*), que tiene el objetivo de desarrollar herramientas *Big Data* que ayuden a los investigadores del área de biomedicina [24] en el estudio de enfermedades como el cáncer.

2.1.2. Desafios

A pesar del interés en proyectos de *Big Data*, hay algunos problemas para los cuales aún se están buscando soluciones adecuadas [25]. La confidencialidad de los datos es uno de estos problemas. Algunas personas y empresas no cuentan con la infraestructura necesaria, como un *cluster* de procesamiento, y por lo tanto utilizan los servicios ofrecidos por un proveedor en la nube. El problema es que no todos los proveedores pueden garantizar las medidas de seguridad adecuadas para mantener los datos seguros. El consumo energético de los sistemas de cómputo es otro problema que ha sido un tema de interés que cuenta con una gran cantidad de investigaciones realizadas al respecto. No obstante, el aumento en la cantidad de datos necesita de más equipos de cómputo para realizar el procesamiento en paralelo para poder hacerlo en un tiempo razonable pero esto se traduce en un mayor consumo energético.

Proyectos *Big Data* son en general necesariamente interdisciplinarios. Se requieren

personas con: i) conocimiento del área de aplicación, es decir del problema a resolver; ii) experiencia en el uso de análisis de datos relevantes al problema; y iii) experiencia en el uso de las herramientas de software *Big Data*. Además, debe considerarse también la integración de las herramientas *Big Data* con la infraestructura the tecnologias de la información existente.

El capítulo continua con una presentación de los diferentes tipos de análisis, herramientas de *software* y productos relacionados con *Big Data*. En el capítulo 3 presentamos tres aplicaciones *Big Data*, los tipos de análisis que realizan, y con qué propósito.

2.2. Tipos de Análisis

El análisis de datos consiste en general en la búsqueda de información útil en forma de patrones que se repiten en un conjunto de datos grande. Por ejemplo, en los negocios el análisis de transacciones (recibos de ventas de productos) ayuda a determinar qué artículos se deberían tener en existencia, porque se ha detectado que están en demanda, con el fin de incrementar las ventas. Este tipo de análisis es llamado minería de datos.

La minería de datos utiliza principalmente tres herramientas: estadística, inteligencia artificial y sistemas de bases de datos [26].

La estadística proporciona técnicas muy bien definidas y estudiadas que permiten lograr un entendimiento básico de los datos y de las relaciones existentes entre las variables analizadas con el fin de prepararlas para su análisis. Estas técnicas permiten organizar los datos, tratar valores ausentes (*missing*), identificar valores atípicos (*outliers*), y realizar transformaciones como normalización, linealización, entre otras. Este tipo de análisis se llama Análisis Exploratorio de Datos o *EDA* (del inglés *Exploratory Data Analysis*). *EDA* es el paso inicial de todos los análisis como regresión, agrupamiento y clasificación.

La inteligencia artificial utiliza heurísticas que contribuyen al procesamiento de información basándose en modelos que simulan el razonamiento humano. Una de

las técnicas de la inteligencia artificial más utilizada por la minería de datos es el aprendizaje de máquina (o en inglés *machine learning*). Por ejemplo, el aprendizaje de máquina es utilizado para la solución de problemas de clasificación, en donde el sistema de aprendizaje trata de etiquetar cada dato con una categoría determinada. La base de conocimiento del sistema está formada por datos etiquetados anteriores. Este tipo de aprendizaje puede llegar a ser muy útil en problemas de bioinformática [5].

Los sistemas de base de datos proporcionan el soporte de acceso, almacenamiento y consulta de los datos que serán analizados.

La minería de datos incluye también la utilización de Procesamiento Analítico En Línea u *OLAP* (del inglés *On-Line Analytical Processing*) y procesos de Recuperación de Información o *IR* (del inglés *Information Retrieval*) [3, 27]. *OLAP* ha sido ampliamente utilizado por las empresas para la toma de mejores decisiones de negocios. Consiste en realizar consultas *SQL* a bases de datos estructuradas alojadas en almacenes de datos (del inglés *datawarehouse*), los cuales son repositorios centrales con información de múltiples fuentes incluyendo bases de datos transaccionales.

IR se refiere a la extracción de información mediante el uso de consultas realizadas sobre datos no estructurados tal como texto o imágenes. Estas consultas pueden ser realizadas sobre documentos o información encontrada en la *Web*. *IR* involucra análisis, clasificación y creación de índices de documentos en base a palabras clave con el propósito de encontrar documentos, o información en documentos, que sea de interés para el usuario.

En lo que sigue de está sección presentamos una despcripción breve de las principales categorias de análisis de datos usados en aplicaciones *Big Data*. Estas son: regresión, clasificación, agrupamiento, detección de anomalías y análisis de sentimiento.

2.2.1. Análisis de regresión

El análisis de regresión es un procedimiento estadístico que estudia la relación funcional entre dos o más variables. El análisis de regresión ayuda a entender cómo el valor de una variable cambia cuando otras son cambiadas, y es usado en áreas tan

distintas como las ingenierías, la física, ciencias económicas, ciencias biológicas y de la salud, ciencias sociales y en otras más [28].

El análisis de regresión permite:

- Investigar si existe una asociación entre dos o más variables
- Determinar la fuerza de la asociación a través de una medida denominada coeficiente de correlación
- Estudiar la forma de la relación. El análisis de regresión permite crear un modelo utilizando los datos disponibles. A partir de este modelo es posible predecir el valor de una variable a partir de la otra

El modelo es una función matemática que describe la relación entre la variable dependiente (Y_t) y la o las variables independientes X_p.

Se han desarrollado varios tipos de análisis de regresión, los cuales se pueden clasificar según diversos criterios tal como el tipo de modelo obtenido o el número de variables independientes. Consideranto el tipo de modelo obtenido se clasifican en:

- Regresión lineal: Cuando Y_t es una función lineal, es decir, el máximo exponente es uno.
- Regresión no lineal: Cuando Y_t no es una función lineal, es decir, tiene funciones trigonométricas o alguno de sus exponentes es mayor que uno.

Considerando el número de variables el análisis de regresión se clasifica en:

- Regresión simple: Cuando la variable Y_t depende únicamente de una única variable X.
- Regresión múltiple: Cuando la variable Y_t depende de varias variables ($X1$, $X2$, ..., Xn)

Por ejemplo, la regresión lineal (múltiple) utiliza una sola variable dependiente Y, una (o más) variables independientes X_p y un término aleatorio ε. Este modelo puede ser expresado como:

$$Y_t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

Donde:

Y_t: variable dependiente.

X_1, X_2, $\cdots$, X_p : variables independientes.

β_0,β_1,β_2,$\cdots$,β_p: son parámetros que miden la influencia que las variables independientes tienen sobre la dependiente.

A partir del uso de análisis de regresión lineal fue posible determinar que el tabaquismo es uno de los factores que aumenta la mortalidad por cáncer pulmonar [29]. Los investigadores que hicieron este estudio trataron de analizar una gran cantidad de variables, como estado socioeconómico, para asegurarse que los efectos de mortalidad por tabaquismo no sean un efecto de su educación o posición económica. Sin embargo fue imposible incluir todas las variables involucradas ya que este estudio fue realizado antes de la llegada de *Big Data*, por lo que este resultado es cuestionado por algunos investigadores, que piensan que algún gen o algún otro factor podría aumentar la mortalidad y aumentar la propensión a adquirir enfermedades relacionadas con el consumo de tabaco.

2.2.2. Análisis de clasificación

La clasificación consiste en asignar una clase o categoría a datos u objetos en base a sus atributos (características) usando un modelo creado con datos previamente clasificados. Ver figura 2.1.

Figura 2.1: Diagrama del funcionamiento de un modelo clasificador.

Los modelos de clasificación, o clasificadores, pueden ser utilizados para distinguir los objetos de diferentes clases. Por ejemplo, los biólogos usan modelos descriptivos (basados en un conjunto de características) de los seres vivos para clasificar otros seres vivos encontrados. Los modelos de clasificación también son utilizados para predecir valores nominales desconocidos, en la sección 3.1.3 veremos un ejemplo de su utilidad.

Los clasificadores pueden ser creados mediante técnicas tales como árboles de decisión, redes neuronales, máquinas de soporte vectorial o *SVM* (del inglés *Support Vector Machines*), entre otras [30]. Cada una de estas técnicas son algoritmos de aprendizaje de máquina que identifican el modelo, una función f, que mejor se adapta al valor de los atributos de los objetos contenidos en un conjunto de datos previamente clasificados conocido como *conjunto de entrenamiento*.

La manera más utilizada para crear clasificadores es mediante el uso de árboles de decision, como sigue. Primero se divide en dos o más partes al conjunto de entrenamiento utilizando alguno de sus atributos. Cada uno de los subconjuntos resultantes representa una rama en el árbol que se debe volver a dividir utilizando algún otro atributo. El proceso se repite hasta que sólo haya datos de la misma clase en cada rama del árbol.

La elección del atributo que se utiliza para realizar una división se puede llevar a cabo mediante el cálculo de la entropía o del índice Gini. El cálculo de estos dos índices se realiza mediante las siguientes ecuaciones:

$$Entropia(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

Donde:

$p(i|t)$ representa la fracción de registros pertenecientes a la clase i en el nodo t

c es el número de clases totales

Además del conjunto de entrenamiento se suele utilizar otro conjunto de datos conocido como *conjunto de prueba* que se utiliza para determinar la precisión del cla-

sificador. Los objetos del conjunto de prueba se encuentran previamente clasificados. Para calcular la precisión del clasificador se compara la clase de todos los objetos del conjunto de prueba y se compara con la clase asignada por el clasificador. Se lleva un conteo de los objetos clasificados correcta e incorretamente y se utiliza la siguiente ecaución:

$$p = \frac{nc}{ni}$$

Donde:

p es la presición

nc es el número de clases asignadas correctamente por el clasificador

ni es el número de clases asignadas incorrectamente por el clasificador

La clasificación es uno los problemas más importantes en la minería de datos y es usada en una amplia gama de aplicaciones. Por ejemplo, los bancos cuentan con información sobre el comportamiento de pago de sus aspirantes a nuevos créditos. Combinando esta información financiera con datos como sexo, edad, ingreso, *etc.*, es posible desarrollar un sistema para clasificar a clientes nuevos como clientes buenos o malos con el fin de estimar el nivel de riesgo para el banco [30].

2.2.3. Análisis de agrupamiento (*clustering*)

El análisis de agrupamiento (del inglés *clustering*) consiste en agrupar objetos que son similares considerando uno o más de sus atributos. El objetivo es que los objetos de un mismo grupo sean similiares o estén relacionados de alguna manera y sean diferentes de los objetos de otros grupos.

En muchos casos, el análisis de agrupamiento es sólo un punto de inicio que permite entender los datos que se están analizando. El análisis de agrupamiento es ampliamente utilizado en ciencias sociales, ingenierías, ciencias biológicas y muchas otras. Por ejemplo, los biólogos han estudiado la taxonomía de los seres vivos, es decir, los han dividido en: reino, filum, clase, orden, familia, género y especie. El análisis de agrupamiento permite crear sistemas de identificación taxonómica automatizados.

Los tipos más comunes de agrupamiento son: jerárquico, particional, exclusivo, borroso, completo, parcial y algunas combinaciones de estos tipos. Un agrupamiento jerárquico permite tener grupos dentro de otros grupos, es decir, grupos anidados. Un agrupamiento particional consiste en dividir los datos en conjuntos sin elementos comunes. En un agrupamiento exclusivo los objetos pertenecen a sólo un grupo. Un agrupamiento borroso es aquél en el que todos los elementos pertenecen a todos los grupos, para esto, se establece un peso para cada elemento que puede tener valores entre 0 (no pertenece) y 1 (pertenece) que indican qué tanto pertenece al grupo. En el agrupamiento completo todos los objetos son asignados a un grupo, mientras que en el parcial puede haber objetos que queden sin ser asignados a un grupo.

De manera general, los algoritmos de agrupamiento determinan un centroide para cada grupo solicitado y asignan los objetos al centroide que es más similar (cercano). Hay muchas medidas que permiten cuantificar el nivel de similaridad de los objetos y la elección de ésta depende de las características específicas de cada conjunto de datos, por ejemplo, se puede utilizar la distancia euclidiana o la distancia de Manhattan [30] cuando los datos son puntos en el espacio, o bien la *similitud coseno* cuando los datos son vectores.

Uno de los algoritmos más utilizados por su facilidad de implementación es *K-means*, el cual es un algoritmo de agrupamiento particional completo. Se puede utilizar la media aritmética, como medida de similaridad de los datos, que componen a un grupo. En *K-means*, un centroide no necesariamente tiene que ser un dato existente en el conjuto, puede ser sólo un punto en el espacio.

K-means funciona de la siguiente manera: Primero se seleccionan K datos del conjunto al azar, los cuales son considerados los centroides iniciales. Segundo, los datos del conjunto (incluyendo los centroides iniciales) se asignan al centroide más cercano. Después se recalculan los centroides, calculando, para cada uno la media aritmética de todos los valores que fueron asignados al mismo. Se repite este cálculo hasta que los centroides no cambien.

Uno de los problemas que puede tener *K-means* es la existencia de grupos vacíos.

Es decir, al momento de elegir los K centroides iniciales de manera aleatoria es posible que ningún objeto del conjunto se asigne a un centroide por estar muy lejano (anomalía) — todos los datos se asignan a los otros centroides. Si esto sucede lo que se debe hacer es reemplazar ese centroide por el dato más diferente (lejano) asignado a otro centroide y repetir el proceso de asignación.

El agrupamiento es usado en una amplia gama de aplicaciones. Por ejemplo, algunas empresas lo utilizan para realizar segmentaciones de mercado que les permita identificar los tipos de clientes existentes en una zona geográfica [30].

2.2.4. Análisis de detección de anomalías

Una anomalía es un dato u objeto dentro de un conjunto que es significativamente diferente del resto de los objetos. Estos objetos son conocidos como *outliers*. La detección de anomalías intenta encontrar los outliers existentes en un conjunto de datos. [30].

Identificar anomalías en los datos es importante porque éstas pueden indicar errores en la obtención de datos, por ejemplo un sensor defectuoso, o bien en algunos casos puede indicar variaciones en algún fenómeno que puede ser de interés para los científicos. Por ejemplo, los análisis de detección de anomalías son ampliamente utilizados por los astrónomos ya que les ha permitido detectar algunos planetas y estrellas desconocidas [31].

Los métodos para la detección de anomalías más comunes utilizan técnicas basadas en:

- Modelos: Utilizan técnicas estadísticas, generalmente se necesita conocer la distribución de los datos para calcular la probabilidad de que un dato sea un *outlier*.
- Proximidad: Se basan en el manejo de distancias entre datos. Si la distancia del un objeto respecto a los demás excede ciertos parámetros el objeto es considerado como un *outlier*.

- Densidad: Se basa en la estimación de densidad de los datos. Es decir, los datos se grafican y se detectan las regiones en la gráfica con baja densidad de objetos. Los objetos que se encuentran en zonas de baja densidad y que se encuentran más alejados de sus vecinos se consideran *outliers*.

Los *outliers* son tratados como ruido o error en muchos de los casos, tal y como sucede en los algoritmos de agrupamiento, o bien son eliminados en el análisis exploratorio de datos. No obstante, los *outliers* pueden ser de utilidad en algunos ámbitos. Por ejemplo, para propósitos de detección de fraude son una herramienta valiosa en la búsqueda de comportamientos atípicos [32].

2.2.5. Análisis de sentimiento

El análisis de sentimiento es el estudio computacional de la opinión de las personas con el fin de determinar sus actitudes y emociones ante ciertos temas o eventos. El objetivo es identificar el sentir de las personas a través de sus opiniones, y clasificarlas de acuerdo a su polaridad [33]. Una opinión puede ser clasificada como positiva o negativa. Los tipos más importantes de análisis de sentimientos son los siguientes (otros son descritos en [33]):

- Clasificación de sentimiento. Realiza una clasificación de un conjunto de opiniones en tres categorías: positivas, negativas o neutrales. Puede ser una tarea compleja cuando las opiniones se encuentran en múltiples idiomas o provienen de varios dominios, como biología, sociología, *etc.*

- Clasificación de subjetividad. Determina si una oración es subjetiva u objetiva. Una oración objetiva contiene información imparcial, mientras que una oración subjetiva contiene información de carácter personal como opiniones.

- Resumen de opinión. Permite extraer las características principales que son compartidas por uno o más documentos y el sentimiento acerca de estas características.

- Recuperación de opinión. Permite extraer documentos que expresan cierta opinión sobre la consulta realizada.

El análisis de sentimiento tiene varios enfoques, el basado en lexicón y el basado en *corpus*.

El enfoque basado en lexicón utiliza una colección de términos conocidos, frases y hasta regionalismos. Este enfoque también es conocido como enfoque basado en diccionarios y utiliza un conjunto inicial de términos que generalmente son recolectados y anotados de manera manual con una categoría (sentimiento) positiva, negativa o neutral. Este conjunto inicial crece al incluir manualmente sinónimos y antónimos de las palabras contenidas. Este diccionario es conocido como bolsa de palabras en la literatura [33]. La principal desventaja del uso de bolsas de palabras es la dificultad de procesar textos con información específica de un determinado contexto o dominio de información, ya que las bolsas de palabras tienden a ser muy generales o muy específicas.

El enfoque basado en *corpus* está basado en el uso de diccionarios para un dominio o contexto en particular. Estos diccionarios pueden ser generados automáticamente, por computadora, a partir de un conjunto de términos semillas proporcionados por los usuarios, y posteriormente son extendidos con la búsqueda de palabras relacionadas a estos términos. Se utilizan métodos estadísticos para determinar qué términos adicionales se agregarán. Cada término tiene también un valor de sentimiento positivo, negativo o neutral.

Se han creado algunos diccionarios o bolsas de palabras que están disponibles en Internet para ser utilizadas sin costo. Por ejemplo, *SentiWordNet*, una extención del diccionario *WordNet*, el cual es una base léxica muy grande de términos en inglés. Este diccionario contiene sustantivos, verbos, adjetivos y adverbios que están agrupados en categorías llamandas *synsets*. Cada una de las palabras agrupadas en el *synsets* tienen un significado similar, diferente al significado de las palabras agrupadas en *synsets* diferentes.

Para determinar la polaridad de una oración, cada una de sus palabras se debe buscar en el diccionario para obtener su valor de sentimiento. El valor individual de cada palabra se utiliza para calcular el valor de la oración. Algunos algoritmos avanzados calculan el valor utilizando conjuntos de palabras y algunos otros hasta el orden de las mismas [33].

El análisis de sentimientos es relativamente nuevo y ha tenido mucha aplicación dentro de los negocios ya que les permite conocer de manera general la opinión sobre un producto o servicio al analizar los comentarios que los clientes publican en redes sociales o en los comentarios de las tiendas en línea. El análisis de sentimiento evita que personas tengan que invertir una gran cantidad de tiempo en leer los comentarios publicados.

2.3. Herramientas

A continuación se describen algunas de las herramientas más utilizadas para realizar tareas de análisis de *Big Data*. Todas las herramientas mencionadas son de código abierto y pueden ser utilizadas sin ningún costo. Estas herramientas son muy poderosas pero han sido pensadas para ser utilizadas por programadores o personas con conocimientos técnicos avanzados.

2.3.1. *MapReduce / Hadoop*

MapReduce es un modelo de programación y ambiente de ejecución desarrollado por la empresa *Google* para procesar grandes cantidades de información de manera paralela utilizando un *cluster* de computadoras de propósito general. Fue propuesto en 2004 por Jeffrey Dean y Sanjay Ghemawat, investigadores de la empresa *Google* [34].

Un programa *MapReduce* consiste de al menos una función *map* y una función *reduce*. Estas funciones son secuenciales y el ambiente *MapReduce* se encarga de replicarlas en múltiples nodos y ejecutarlas en paralelo de manera transparente al programador. El ambiente *MapReduce* cuenta con balance de carga y tolerancia a fallos

también de manera transparente para el programador.

Las funciones *map* y *reduce* se ejecutan de manera paralela y distribuida en un *cluster*. El ambiente *MapReduce* replica las funciones *map* y *reduce* en los nodos del *cluster*, de tal manera que las réplicas de cada función se ejecutan al mismo tiempo en nodos distintos (ver figura 2.2). Los datos de entrada a las réplicas de la función *map* se encuentran almacenados en un archivo paralelo y distribuido: datos distintos se almacenan en nodos distintos para que sean procesados simultáneamente y así reducir el tiempo total de acceso a datos. Los datos de salida de las réplicas de la función *reduce* también se escriben en el sistema de archivos paralelo y distribuido.

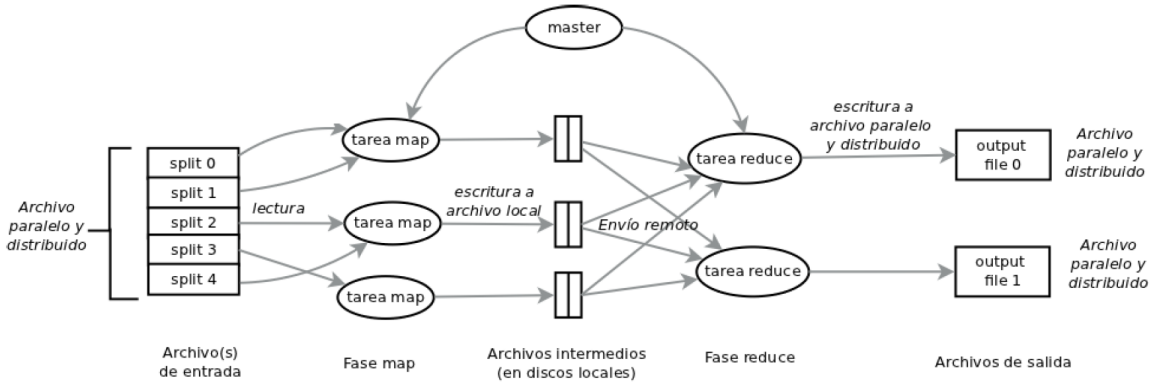

Figura 2.2: Ambiente de ejecución de *MapReduce*

El siguiente ejemplo muestra el pseudocódigo de un programa realizado con el modelo de programación *MapReduce*. El programa cuenta el número de veces que aparece cada palabra en uno o más archivos. La función *map* lee los datos en los archivos, separa cada palabra e imprime una línea compuesta por cada palabra y un 1. El ambiente *MapReduce* organiza (por medio de particionamiento) las salidas de todas las funciones *map* de tal manera que todos los unos de una palabra conforman una lista. Entonces se ejecuta la función *reduce*, una vez por cada palabra y se le proporciona la lista correspondientes de unos. La función *reduce* suma los unos contenidos en la lista e imprime el resultado.

```
1  // clave: desplazamiento dentro del archivo
2  // valor: línea de archivo a procesar
3  map (String clave, String valor)
4  {
5      linea = valor;
6      palabras[] = ObtenerPalabrasDeLaLinea(linea);
7
8      i = 0 ;
9      while (i < palabras.length)
10     {
11         palabra = palabras[ i ];
12         emit(palabra, 1);
13         i++;
14     }
15 }
16
17 // clave: palabra
18 // valores: lista de unos.
19 reduce (String clave, Iterator valores)
20 {
21     sum = 0 ;
22
23     for each v in valores
24     {
25         sum += v ;
26     }
27
28     emit (clave, sum) ;
29 }
```

Uno de los usos más importantes de *MapReduce* dentro de *Google* es en su motor de búsqueda en la *Web* [34]. Cuando un usuario realiza una búsqueda, un sistema de indexado (varios índices) es usado para identificar las páginas *Web* y documentos que contienen los términos de búsqueda dados por el usuario. Tales páginas y documentos

han sido obtenidos por un *crawler* con anterioridad [35]. El sistema de indexado es construido utilizando *MapReduce* cada 24 horas.

Hadoop es una implementación de código abierto de *MapReduce* que fue creada en el año 2005 por Doug Cutting quien posteriormente se uniría a la empresa *Yahoo!*, donde recibió apoyo para terminar el desarrollo de *Hadoop*.

Hadoop ha sido diseñado para ser escalable en cuanto al número de nodos que soporta. Puede ser ejecutado en una sola computadora o en *cluster* de miles de ellas. Cada nodo cuenta con su propio sistema de almacenamiento y procesamiento local. En el año 2008 *Yahoo!* utilizó un *cluster* constituido por 10,000 nodos *Hadoop* para crear su propio motor de búsqueda en la *Web* [35].

Actualmente *Hadoop* es un proyecto de la Fundación Apache y cuenta con cuatro módulos principales:

- *Hadoop Common*. Son un conjunto de utilidades que darán soporte a los demás módulos de *Hadoop*.
- *Hadoop Distributed File System* (*HDFS*). Es el sistema de archivos distribuidos, escalabe y portable de *Hadoop*, es la versión libre del sistema de archivos Google File System.
- *Hadoop YARN*. Es un framework utilizado para la calendarización de procesos y administración de recursos del *cluster*.
- *Hadoop Mapreduce*. Es un sistema construido sobre *YARN* que permite la ejecución en paralelo de programas *MapReduce* con grandes conjuntos de datos.

Se han creado muchos proyectos de software que utilizan *Hadoop* tales como *Avro*, *Cassandra*, *Hive*, *Mohout*, y muchos más. En las secciones siguientes de éste capítulo se describirá algunos de estos proyectos. *Hadoop* es utilizado por muchas empresas a nivel mundial tal como *Facebook*, el *New York Times* y *Last.fm*.

Hadoop está escrito en *Java* y puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 2.7 y fue liberada en abril de 2015.

2.3.2. *Spark*

Spark es un framework de código abierto para la creación de aplicaciones distribuidas utilizando el modelo de programación *MapReduce*. A diferencia de *Hadoop* que escribe los resultados intermedios de las funciones *map* y *reduce* al disco, *Spark* cuenta con primitivas que le permiten realizar todas las operaciones en la memoria de la computadora sin la necesidad de realizar ninguna escritura de resultados intermedios en el disco de la computadora.

Dependiendo del tipo de problema y de los datos a procesar, *Spark* puede ser hasta 100 veces más rápido que *Hadoop* [36]. Un requerimiento indispensable para lograr este desempeño es que los datos completos deben caber en la memoria de las computadoras utilizadas, sin lo cual el rendimiento tiende a deteriorase mucho.

Spark se desarrolló en el año 2009 por Matei Zaharia en la Universidad de California en Berkeley y actualmente es un proyecto de la Fundación Apache que cuenta con los siguientes componentes:

- *Spark Core*. Es el motor de Spark y su función es proveer calendarización de procesos y funciones básicas de entrada y salida a los demás componentes del sistema.
- Spark SQL. Este componente ayuda a gestionar una abstracción de datos llamada *DataFrame*, que provee soporte para el procesamiento de datos estructurados y semiestructurados que se encuentran en memoria mediante consultas en leguaje SQL.
- *Spark Streaming*. Es un módulo que brinda las funciones necesarias que le permiten a Spark procesar flujos de datos.
- *MLib*. Es un framework para la implementación de algoritmos distribuidos de aprendizaje de máquina.
- *GraphX*. Este módulo permite construir y transformar grafos, incluye una librería con implementaciones de algunos algoritmos de uso general.

Spark es escalable en cuanto al número de nodos que soporta y puede ser instalado en una computadora o en un cluster miles de computadoras. *Spark* puede ser utilizado junto con otros sistemas como *Hadoop*, *Cassandra*, *Hive*, entre muchos otros.

Este sistema es uno de los más activos en la Fundación Apache, en junio de 2015 contaba con más de 570 desarrolladores.

A finales del año 2014, un cluster Spark conformado por 206 nodos virtualizados en la nube de *Amazon* pudo ordenar 100 terabytes de datos en sólo 23 minutos, rompiendo el record anterior de 72 minutos de un cluster *Hadoop* con 2100 nodos físicos.

Los módulos de *Spark* están escritos en *Scala*, *Java* y *Phyton*. *Spark* puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 1.4 que fue liberada en junio de 2015.

2.3.3. *Mahout*

Mahout es una biblioteca con algoritmos de minería de datos y aprendizaje de máquina que han sido implementados utilizando el modelo *MapReduce*. La mayoría de los algoritmos pueden ser ejecutados en *Hadoop* y algunos otros en *Spark*. El desarrollo de *Mahout* aún continúa, muchos algoritmos populares aún no se incluyen pero su número crece continuamente [12].

El desarrollo de *Mahout* fue iniciado por alguno de los desarrolladores del proyecto Apache Lucene, el cual es un buscador de código abierto. Esto debido a que necesitaban una plataforma robusta de aprendizaje de máquina para ser usada dentro de *Lucene*. Desde sus inicios, *Mahout* ha sido un proyecto de la Fundación Apache.

Mahout está escrito en *Java* y *Scala*. Puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 0.10 y fue liberada en mayo de 2015. En esta versión se han implementado más de 40 algoritmos de diversos tipos tales como clasificación, agrupamiento, entre otros. Una lista completa de los algoritmos implementados puede consultarse en [37].

2.3.4. *Hive*

Hive es un sistema de base de datos implementado sobre *Hadoop*. Hive cuenta con su propio lenguaje de consultas llamado *HiveQL* el cual es similar a *SQL* pero con la posibilidad de incorporar código *MapReduce* embebido dentro de las consultas.

Hive hace posible analizar grandes conjuntos de datos almacenados en *HDFS* y también es compatible con el sistema de archivos *S3* de *Amazon*. En las últimas versiones *Hive* cuenta con manejo de transacciones pero tiene poco soporte para realizar consultas anidadas (subconsultas).

Hive fue desarrollado inicialmente por la empresa *Facebook* y actualmente es un proyecto de la Fundación Apache.

Hive está escrito en *Java* y puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 1.1 que fue liberada en marzo de 2015.

2.4. Productos de *Big Data*

A continuación se describen algunos de los productos más utilizados para realizar análisis de *Big Data*. Algunos de estos productos cuentan con versiones de uso gratuito y algunos otros ofrecen versiones comunitarias de código abierto. No obstante la mayoría de las versiones gratuitas cuentan con funcionalidad limitada. Estos productos cuentan con una gran cantidad de análisis, son fáciles de usar y han sido pensados para ser utilizadas por personas con pocos conocimientos técnicos.

2.4.1. *Databricks*

Databricks es una plataforma *Web* en donde los usuarios pueden realizar análisis de *Big Data*. Fue creada en junio de 2014 por la empresa *Databricks Inc*, la cual fue fundada por los creadores de *Spark*. *Databricks* se compone de un cluster de procesamiento (llamado *Databricks Platform*) y de una interfaz gráfica *Web* (llamanda

Databricks Workspace).

Databricks [15] permite a los usuarios crear y administrar un cluster virtual basado en *Spark*, es decir, mediante la interfaz gráfica los usuarios pueden especificar las características del *cluster* de procesamiento que desean utilizar en sus análisis. Se pueden configurar características tales como número de nodos, capacidad de almacenamiento, cantidad de memoria y algunas otras. Dependiendo de las características especificadas es el costo que se deberá pagar. *Databricks* utiliza las nubes de *Amazon*, *Microsoft Azure* y *Google Cloud Platform* para crear el cluster especificado por el usuario.

Databricks tiene integrados varios *frameworks* de análisis de datos compatibles con *Spark* como *Mahout*. *Databricks* permite realizar procesamiento por lotes, procesamiento de consultas interactivas, aprendizaje de máquina y análisis de datos en *stream*.

La interfaz gráfica permite visualizar e interactuar con los resultados de los análisis. Cuenta con un sistema de colaboración que permite compartir los resultados con otros usuarios. También posee un módulo en el que los usuarios pueden crear consultas sobre los datos de resultados utilizando lenguajes como *Python*, *SQL*, y *Scala*.

2.4.2. *RapidMiner*

RapidMiner es una plataforma que provee un entorno para realizar procesos de aprendizaje de máquina, minería de datos y análisis predictivos. Es ampliamente utilizado en el área de la investigación y educación, para la creación rápida de prototipos y en aplicaciones empresariales [16] ya que cuenta con varias herramientas de visualización.

En encuestas realizada por *KDnuggets*, un sitio *Web* popular enfocado a temas de minería de datos, *RapidMiner* resultó ser la plataforma *Big Data* más utilizada entre los suscriptores al sitio [38, 39].

RapidMiner cuenta con 5 versiones: Versión Inicial (Starter), Versión Personal (*Personal*), Versión Profesional (*Professional*), Versión Profesional Plus (*Professional*

Plus) y Versión Empresarial (*Enterprise*). Se puede obtener mayor información de estas versiones en [16].

RapidMiner está escrito en *Java* y puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 6.1 y fue liberada en octubre de 2014.

2.4.3. *Pentaho*

Pentaho es una plataforma *Big Data* especialmente diseñada para realizar Inteligencia de Negocios o *BI* (del inglés *Business Intelligence*). Ofrece herramientas para realizar procesos de Extracción, Transformación y Carga o ETL (del inglés *Extract, Transform and Load*), creación de reportes y minería de datos.

Pentaho cuenta con muchos módulos y herramientas, algunas de ellas son de código abierto y pueden ser descargadas gratuitamente de Internet pero algunas otras sólo están disponibles para suscriptores de pago. Algunos de los módulos más importantes de *Pentaho* se describen a continuación:

Plataforma *BI*. Es el motor de *Pentaho* y permite interconectar a los demás módulos y herramientas. Cuenta con una interfaz *Web*, con la que los usuarios pueden crear y visualizar reportes de los datos que previamente han sido configurados en el sistema.

Mondrian. Es una herramienta para realizar Procesamiento Analítico en Línea u OLAP (del inglés *On-Line Analytical Processing*) en la que es posible realizar consultas utilizando el lenguaje *MDX*. Esta herramienta puede ser utilizada de manera individual aunque está diseñada para ser utilizada dentro del entorno de *Pentaho*.

Kettle. Esta herramienta permite realizar procesos de *ETL* mediante una interfaz gráfica arrastrando e interconectando íconos para crear flujos de trabajo. Esta herramienta puede ser usada en una sola computadora o configurada para ser utilizada en un cluster de computadoras.

Pentaho Weka. Es una herramienta construida sobre un *framework* de aprendizaje de máquina y minería de datos llamado *Weka* y que contiene una gran cantidad de algoritmos de regresión, clasificación y visualización.

Pentaho cuenta con una versión Comunitaria (*Community*) y una versión Empresarial (*Enterprise*). La versión Empresarial provee soporte técnico, acceso prioritario a nuevas versiones y soporte de módulos especializados que no disponibles en la versión Comunitaria. Es posible utilizar la versión Empresarial mediante el pago de una subscripción anual cuyo costo depende de las características de cada empresa.

Pentaho está escrito en *Java* y puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 5.0 y fue liberada en abril de 2014.

2.4.4. *KNIME*

KNIME es una plataforma libre y de código abierto que integra varios componentes para el aprendizaje de máquina y minería de datos. Incluye una interfaz visual, en donde los usuarios pueden arrastrar e interconectar íconos para realizar procesos *ETL*.

KNIME se empezó a desarrollar en el año 2004 por la Universidad de Konstanz en Alemania con un grupo de desarrolladores encabezados por Michael Berthold. El objetivo era crear un software de minería de datos para ser utilizado en la industria farmacéutica. En el año 2006 el desarrollo de *KNIME* fue terminado y se liberó la primer versión.

KNIME cuenta con una *API* pública que puede ser fácilmente extendida según las necesidades del usuario y puede ser utilizada en conjunto con *Hadoop* y *Hive*.

KNIME está escrito en *Java* y puede ser ejecutado en el sistema operativo *Linux*, *MacOS* y *Windows*. La última versión estable es la 2.11 y fue liberada en enero de 2014.

2.5. Resumen

Este capítulo presentó una introducción a los conceptos de *Big Data*, algunos de los análisis que se realizan, las herramientas que se utilizan para programar análisis de *Big Data* y algunos los productos más populares existentes en el mercado.

Un análisis de *Big Data* es el proceso de examinar datos con una variedad de tipos u orígenes, con el fin de descubrir patrones ocultos, correlaciones desconocidas y otra información útil. El objetivo principal de los análisis de *Big Data* es ayudar a la toma de mejores decisiones. Los análisis de *Big Data* hacen posible analizar grandes volúmenes de datos transaccionales, así como otros orígenes de datos que no han sido explotados. Los tipos de análisis que se pueden realizar son: regresión, clasificación, agrupamiento, detección de anomalías y análisis de sentimiento. También implica *OLAP* (*On-line Analytical Processing*) e *IR* (*information retrieval*).

MapReduce es la herramienta más importante para realizar análisis de *Big Data*. Es un modelo de programación y ambiente de ejecución desarrollado por *Google* para procesar grandes cantidades de información de manera paralela utilizando un *cluster* de computadoras conformado por nodos de propósito general. *MapReduce* cuenta con balanceo de carga y tolerancia a fallos transparentes para el programador.

Hadoop es una implementación de código abierto de *MapReduce*.

Spark es otro *framework* de código abierto para la creación de aplicaciones distribuidas utilizando el modelo de programación *MapReduce*. *Spark* cuenta con primitivas que le permiten realizar todas las operaciones en la memoria de la computadora y dependiendo del tipo de problema puede ser hasta 100 veces más rápido que *Hadoop* [36].

Se han creado varios productos que hacen uso de las herramientas para *Big Data* tales como *Hadoop* o *Spark*. La ventaja de estos productos es que los usuarios no deben realizar ningún programa para poder utilizarlos y no es necesario que tengan conocimientos avanzados en computación. La mayoría de los productos existentes son comerciales pero tienen versiones limitadas de uso gratuito. Algunos de estas productos son: *RapidMiner*, *Pentaho* y *KNIME*.

Capítulo 3

Aplicaciones de *Big Data*

Hay muchas aplicaciones para *Big Data*. Algunas de estas aplicaciones son antiguas, realizadas antes del uso del término *Big Data*, pero ahora se están resolviendo utilizando técnicas de *Big Data* con lo que se ha conseguido ahorro en tiempo y dinero. En este capítulo se abordan algunas de las aplicaciones *Big Data* en los negocios y en las ciencias.

3.1. *Business Intelligence* (*BI*)

BI era muy costosa. Sólo grandes empresas podían costearla. Típicamente, los datos que se deben analizar, para poder identificar patrones significativos útiles, deben ser muchos, son de distintos tipos, y provienen de múltiples fuentes; por lo que su procesamiento es complejo, lento y muy costoso con los métodos, herramientas y plataformas usadas antes de la llegada de las herramientas de *Big Data*. En particular, las plataformas eran computadoras paralelas de alto desempeño, pero no de propósito general (como los *clusters* usados con *Big Data*), y por lo tanto costosas. Las herramientas *Big Data* han hecho posible que empresas grandes y pequeñas puedan analizar datos de sus actividades para optimizarlas.

BI ofrece muchos beneficios y las empresas que la han adoptado tienden a invertir en la creación de más y mejores herramientas de análisis. Por ejemplo, *Social Genome*

es una herramienta creada por *Walmart* para realizar investigaciones de mercado en base a los comentarios publicados en redes sociales [40]. *Social Genome* utiliza un entorno de ejecución basado en *MapReduce/Hadoop* que ha sido modificado para el procesamiento de datos en tiempo real. *Social Genome* es ahora libre y abierta, pero inicialmente fue de uso privado [40].

En el ámbito de los negocios, se estima que la cantidad de información almacenada a nivel mundial se duplica cada 14 meses [41]. *WalMart* cuenta con más de 11,000 tiendas alrededor del mundo y realiza más de 267 millones de transacciones (ventas) al día [42]. Se calcula que los datos almacenados a la fecha sólo por *WalMart* equivalen a 2.5 petabytes [43].

Además de datos transaccionales, las empresas también tienen acceso a una enorme cantidad de datos provenientes de redes sociales, en las que se incluye opiniones sobre los productos o servicios ofrecidos por la empresa y hasta por sus competidores. Las empresas también almacenan gran cantidad de información relacionada con sus procesos de producción y administración. El proceso de extracción, almacenamiento y análisis de datos de las distintas actividades de una empresa para encontrar información útil en la toma de mejores decisiones se conoce como Inteligencia de Negocios o *BI* (del inglés *Business Intelligence*). Sus propósitos incluyen: la investigación de mercados, la creación de modelos para la reducción de riesgos, la optimización de procesos de producción, la identificación de oportunidades de crecimiento, entre otros objetivos [44].

En esta sección vamos a presentar la investigación de mercados.

3.1.1. Investigación de mercados — etapas principales

La investigación de mercados (IM) tiene el propósito de estudiar a los clientes de una empresa para incrementar las ventas por medio de mejorar sus productos y servicios considerando la opinión de los clientes. Anteriormente, esto se lograba por medio de encuestas de satisfacción lentas y propensas a errores por ser realizadas por encuestadores humanos por teléfono y de manera presencial. Hoy en día se realiza

principalmente analizando información en *sistemas de recomendación* y la opinión que clientes publican en redes sociales sobre los productos de interés, como es descrito más adelante.

La investigación de mercados consiste de cuatro etapas: conocimiento de clientes, segmentación de clientes, creación de campañas publicitarias, y medición de la efectividad de las campañas publicitarias. En esta sección presentamos una descripción breve de cada etapa y en las secciones subsecuentes presentamos como se realiza cada etapa.

El conocimiento de clientes consiste en asignar atributos o etiquetas a cada cliente en base a información de: i) los tipos de productos que compra; ii) los comentarios o calificación que asigna a productos que ha comprado; iii) la calificación que da a comentarios de otros clientes en el sitio Web de la tienda o en las redes sociales como twitter y facebook. Esta información se obtiene y maneja con un *sistema de recomendación*. Casi todos los sitios web de empresas que han adoptado tecnologías *Big Data* manejan un sistema de recomendación.

La segmentación de clientes es el proceso de identificar grupos de clientes considerando los productos que compran. Se realiza utilizando algoritmos de agrupamiento y clasificación [45]. Los grupos de clientes se identifican por atributos como: nivel económico, etnicidad, edad, sexo, nivel educativo, etc.

Las campañas publicitarias se organizan solo para algunos de los grupos de clientes identificados en la etapa segmentación de clientes, con el propósito de que sean más efectivas. Por el ejemplo, la publicidad dirigida a niños generalmente incluye dibujos animados, mientras que la publicidad dirigida a adultos tiende a ser más seria.

Finalmente, la medición de la efectividad de campañas publicitarias consiste en determinar el incremento de ventas correspondiente. Esto se realiza por medio de algoritmos de regresión que relacionan ventas de productos (fecha, hora, lugar de aplicación, medio de comunicación, etc.) con una o más campañas publicitarias [46]. Por ejemplo, si después de que un cliente compra un producto X, también compra Y de la lista "Clientes que compraron el producto X también compraron Y y Z",

entonces se considera que el sistema de recomendaciones está siendo efectivo.

3.1.2. Conocimiento de clientes

Recuérdese que el conocimiento de clientes incluye el asignar atributos o etiquetas a cada cliente en base a: i) los tipos de productos que compra; ii) los comentarios o calificación que asigna a productos que ha comprado; iii) la calificación que da a comentarios de otros clientes en el sitio Web de la tienda o en las redes sociales como twitter y facebook.

Con tal información, lo que se busca es determinar la opinión, positiva o negativa, que cada cliente tiene de un producto. Este proceso se realiza utilizando análisis de sentimiento. En la Sección 2.2.5 vimos que el análisis de sentimiento consiste en determinar las actitudes y emociones de las personas ante ciertos temas o eventos por las palabras que utilizan al escribir de los mismos.

Es importante resaltar que las opiniones que se utilizan en el análisis de sentimiento no son, en general, del uso exclusivo de las empresas que los obtuvieron. El público en general puede hacer uso de tales opiniones, por medio del *sistema de recomendación* de una empresa, para decidir comprar o no algún producto — analizando las comentarios hechos por clientes que ya compraron el producto. Este comportamiento se conoce como recomendaciones de boca en boca o *WOM* (del inglés *Word-of-Mouth*). Los sistemas de recomendación tienden a atraer la opinión de más clientes, lo que representa más datos que permiten *afinar* las opiniones e identificar mejores productos y servicios [47]. Además, las empresas también utilizan los datos de opiniones para *conocer* (agrupar) a sus clientes y poder ofrecerles otros productos y servicios acordes a sus intereses.

Los sistemas de recomendación son ampliamente utilizados en las tiendas en línea (*eCommerce*) [48], pero son muy útiles también como base de comparación para la compra de productos no en línea.

3.1.3. Segmentación de clientes

Recuérdese que la segmentación de clientes es el proceso de identificar grupos de clientes considerando los productos que compran. Los grupos se identifican utilizando algoritmos de agrupamiento [45] que toman en cuenta atributos (variables) geográficos, demográficos, psicográficos y conductuales. Sin embargo, antes del agrupamiento se utiliza clasificación como parte de un proceso de preparación de los datos como se describe adelante.

Mientras que en el paso anterior, conocimiento de clientes, se determina la opinión de los clientes sobre un producto, en la segmentación de clientes nos interesa conocer "quiénes" son esos clientes, de dónde son, qué gustos tienen, etc. Por ejemplo, las variables geográficas que se manejan incluyen país, región, ciudad, clima; las variables demográficas incluyen edad, género, nivel educativo, religión, cultura y raza, entre otros; las variables psicográficas incluyen personalidad, estilo de vida (actividades de ocio o aficiones, hábitos alimenticios, *etc.*), actitudes e intereses; y las variables conductuales incluyen taza de utilización del producto, fidelidad a la marca, nivel de listo-para-consumir, entre otros.

Es común que no se tenga información completa de todos los clientes. Y si la información faltante en un cliente es utilizada por el algoritmo de agrupamiento, tal cliente no será agrupado correctamente y el agrupamiento será menos eficaz. Algunos datos se pueden deducir en base a otra información. Por ejemplo, el país puede determinarse mediante la dirección *IP* utilizada al momento que un cliente realizo su registro o considerando la *IP* más utilizada para conectarse al sitio web de la empresa.

Si datos faltantes no se pueden determinar utilizando otras fuentes de información, entonces se usa clasificación, como sigue. Se recordará que la clasificación compara datos nuevos con el modelo generado a partir de datos previamente clasificados para asignarles una etiqueta. Así, si no se tiene la edad de un cliente, se le puede asignar una edad aproximada a la de otros clientes que tienen intereses similares.

El agrupamiento de los clientes busca identificar tipos de clientes que tienen atributos comunes, como edad, país, intereses similares, etc. Determinar los tipos de clientes

permite a los analistas de negocios entender las opiniones de los clientes sobre productos, y tomar decisiones para mejorar la calidad de los productos y así incrementar las ventas.

3.1.4. Campañas publicitarias

Recordamos que las campañas publicitarias se organizan solo para algunos de los grupos de clientes identificados en la etapa segmentación de clientes. A estos grupos les llamaremos clientes objetivo; también se les conoce como público objetivo (y en inglés como *target market*).

La publicidad se diseña para ser interesante a los clientes objetivo. Por ejemplo, la publicidad dirigida a niños generalmente incluye dibujos animados, mientras que la publicidad dirigida a adultos tiende a ser más seria. Además, considerando la información de los sistemas de recomendación, y las compras individuales de un cliente, los sistemas *Web* de empresas corren campañas publicitarias en línea que sugieren al cliente, al momento de comprar un producto, otros productos similares o de alguna manera relacionados y que el cliente posiblemente no conocía y que el sistema ha detectado que son o pueden ser de interés a otros clientes dentro del mismo tipo de clientes.

Por ejemplo, la tienda en línea de *Amazon* cuenta con un catálogo de más de 183 millones de productos [49] por lo que cuenta con un sistema de recomendación basado en análisis de clasificación, agrupamiento y filtros colaborativos. El sistema de *Amazon* asocia cada producto comprado por usuario con una lista de productos similares, que se obtiene en función de los productos que hayan sido comprados en un mismo pedido por otros clientes. Los filtros colaborativos son técnicas de filtrado de datos que utilizan los datos de múltiples fuentes, en las que se incluye los datos del usuario, junto con los datos de otros usuarios similares. De esta manera, los filtros colaborativos permiten encontrar información *faltante* en el conjunto de datos original [50].

3.1.5. Evaluación de campañas publicitarias

La medición de la efectividad de campañas publicitarias consiste en determinar el incremento de ventas correspondiente. Esto se realiza por medio de algoritmos de regresión que relacionan información de ventas de productos (fecha, hora, lugar de aplicación, medio de comunicación, etc.) con una o más campañas publicitarias [46]. Por ejemplo, si después de que un cliente compra un producto X, también compra Y de la lista "Clientes que compraron el producto X también compraron Y y Z", entonces se considera el sistema de recomendaciones está siendo efectivo.

3.2. Bioinformática

La bioinformática es una ciencia emergente que utiliza tecnologías de la información para organizar y analizar información biológica, con la finalidad de responder preguntas complejas en biología. Por ejemplo, la bioinformática ha hecho posible la investigación del genoma humano el cual puede ayudar dramáticamente a mejorar las condiciones y calidad de vida, al estudio de enfermedades, a la producción de alimentos genéticamente modificados, entre otros.

A continuación se presentan dos ejemplos del uso de tecnología *Big Data* en la bioinformática.

3.2.1. Secuenciación del DNA

El *DNA* (del inglés *deoxyribonucleic acid*), es una molécula que guarda y transmite de generación en generación toda la información necesaria para el desarrollo de las funciones biológicas de un organismo. Está formado por la unión de dos cadenas construidas con elementos de 4 tipos diferentes llamados nucleótidos (adenina, timina, citosina y guanina). La cantidad y orden de estos nucléotidos se llama código genético o genoma. El genoma humano consta de tres mil millones de pares de nucléotidos.

El estudio del *DNA* ayuda a los científicos a entender enfermedades con la finalidad

de poder crear tratamientos, también les permite buscar similitudes entre diferentes especies y crear organismos modificados genéticamente tal como especies vegetales resistentes a plagas. Sin embargo, la enorme cantidad de información contenida en el *DNA* es muy grande por lo que para su estudio se requiere la ayuda de computadoras.

Para obtener una versión digital del *DNA* los científicos se apoyan en un proceso llamado secuenciación que consiste en extraer y determinar la cantidad de nucleótidos contenidos en una muestra de *DNA*. La secuenciación se realiza en un aparato llamado secuenciador. El proceso de secuenciación consta de 3 etapas: marcado, separación y secuenciación. El *DNA* es muy grande por lo que debe procesarse en fragmentos. Durante la etapa de marcado se determina el tamaño ideal de los fragmentos y se identifica el lugar en donde se debe cortar el DNA. En la etapa de separación se utilizan enzimas especiales que realizan este corte. Finalmente, durante la etapa de secuenciación se determinan los nucleótidos que conforman el fragmento que se está procesando. Sin embargo, los secuenciadores generan una versión desordenada de los nucleótidos por lo que éstos deben ser ordenados mediante el uso de computadoras. Este proceso de ordenamiento se llama alineación. En la figura 3.1 se muestra el resultado del proceso de secuenciación, las letras A, T, G, C representan respectivamente los nucleótidos de adedina, tiamina, guanina y citocina. Los guiones representan la existencia de nucleótidos que no fueron identificados.

Tradicionalmente, el proceso de alineación se realiza mediante el uso de la herramienta llamada *BWA* (del inglés Burrows-Wheeler *Aligner*). No obstante, la enorme cantidad de datos generados durante la secuenciación hace que la alineación sea un proceso lento y costoso. Procesar un solo fragmento de *DNA* utilizando *BWA* requería de más de siete días con un costo de un de un millón de dólares en 2007 [51].

El uso de herramientas *Big Data* logró disminuir considerablemente el costo y tiempo requerido por el proceso de alineación a mil dólares en 2012 [51]. Por ejemplo, la herramienta *BigBWA* es una adaptación de los algoritmos de *BWA* al modelo *MapReduce* [51]. *BigBWA* ha logrado realizar el proceso de alineación en sólo ocho horas utilizando un cluster *Hadoop* conformado por 60 nodos.

Figura 3.1: Contenido del archivo resultante del proceso de secuenciación de DNA

Gracias al uso de herramientas *Big Data*, la secuenciación y alineación del *DNA* se ha echo costeable, tanto que en los siguientes años el Instituto Nacional del Cáncer de Estados Unidos o *NCI* (del inglés *National Cancer Institute*) planea secuenciar un millón de genomas para determinar las variaciones y patrones existentes con el fin de compararlo con el genoma encontrado en células de tumores. Esta es otra aplicación de *Big Data* que se describe a continuación.

3.2.2. Reconocimiento de patrones

El resultado del proceso de secuenciación del *DNA* consiste en uno o varios archivos que en total contienen más de 110GB de datos. Estos archivos contienen la representación en forma de texto de los nucléotidos que conforman al *DNA*. El contenido de uno de estos archivos puede verse en la figura 3.1. La búsqueda de patrones en estos archivos ha permitido identificar, por ejemplo, los genes responsables del desarrollo de algunas enfermedades.

Por ejemplo, los *SNP* (del inglés *Single Nucleotide Polymorphism*) son secuencias de nucleótidos en un mismo *DNA* que varían en solo uno de sus elementos. Actualmen-

te se manejan secuencias *SNP* de hasta 30 nucleótidos (como se describe adelante). Los genetistas han encontrado que estas variaciones pueden afectar a la respuesta de los individuos a enfermedades, bacterias, virus, productos químicos, fármacos, etc. Es por eso que su estudio es muy importante para los genetistas.

La búsqueda de *SNPs* ha sido posible gracias al uso de herramientas *Big Data* ya que implica el procesamiento de al menos un genoma completo. Por ejemplo, *Crossbow* [52] es una herramienta para la búsqueda de *SNP* que se encuentra implementada utilizando *MapReduce*. *Crossbow* utiliza un *cluster* conformado por 320 nodos *Hadoop* virtualizados en la nube de *Amazon*. *Crossbow* es capaz de buscar *SNPs* en el genoma de una persona en sólo tres horas. *CrossBow* es considerada como una de las herramientas más avanzadas en la búsqueda de *SNPs* pero aún es incapaz de procesar secuencias de más de 30 nucléotidos, en un tiempo razonable.

3.2.3. Herramientas para bioinformática

En el área de bioinfomática muy pocas herramientas están diseñadas para ser ejecutadas en paralelo. Las herramientas de *Big Data* son muy poderosas pero aún no han sido adoptadas por la mayoría de los investigadores, debido a que su utilización requiere de conocimientos técnicos avanzados en el área de la computación. Aunque el potencial de las herramientas *Big Data* es reconocido en el entorno académico, pocas acciones se realizan para implementarlas a gran escala [53].

Con el fin de ayudar a la implementación de las herramientas Big Data, muchas empresas líderes en tecnología han anunciado el apoyo al área de bioinformática. Por ejemplo. Dell está donando capacidad de procesamiento no utilizada en sus servidores con el fin de crear el primer software especializado, aprobado por la *FDA* (del inglés *Food and Drug Administration*), para el estudio de cáncer en niños, en especial del tipo neuroblastoma. Este software será creado por el Instituto Trasnacional de Investigación Genómica o TGen (del inglés *Trasnacional Genomics Research Institute*) [54].

Intel está colaborando en el desarrollo de *Nextbio*, un *software* utilizado por los

investigadores del área de ciencias biológicas, con el fin de optimizar el Sistema de Archivos Distribuidos de *Hadoop* y *HBase* [55]. Aunque *Nextbio* es un software privado todas las optimizaciones serán públicas y de código abierto.

Cloudera se ha asociado con el Instituto de Genómica y de Biología Multiescala (*Institute of Genomics and Multiscale Biology*) en Estados Unidos en un esfuerzo por crear una herramienta que, utilizando técnicas de *Big Data*, ayude en el estudio y tratamiento de enfermedades [56]. Se pretende estudiar el genoma de bacterias y el metabolismo de organismos sanos y enfermos.

3.3. Análisis de datos geoespaciales

El término geoespacial comprende un conjunto muy amplio de disciplinas que requieren el uso de tecnologías que permitan almacenar, procesar, visualizar y analizar datos que cuentan con una posición definida dentro de un área en particular. Estos datos se conocen como datos georreferenciados. Entre los sistemas que hacen uso de datos georreferenciados destacan los Sistemas de Información Geográfica o *GIS* (del inglés *Geographic Information System*), las imágenes satelitales y los Sistemas de Posicionamiento Global o *GPS* (del inglés *Global Positioning System*).

El área geoespacial ha tenido retos muy grandes debido al gran volumen de datos generados por los satélites [57]. Estos datos incluyen imágenes satelitales que, en varias áreas de la ciencia, son un método fundamental para la adquisición de datos de la superficie de la Tierra [57]. Por ejemplo, gracias a la alta resolución de las cámaras colocadas en los satélites, en los últimos años se ha logrado mejorar la precisión con la que se determina el grado de deforestación de áreas boscosas. Las imágenes satelitales son usadas también en situaciones de emergencia, donde es frecuente implementar sistemas de monitoreo en tiempo real para realizar estimaciones de riesgo [57].

A continuación se describe brevemente cómo el análisis de los datos provenientes de los sistemas GPS ayudan a modelar el comportamiento del tráfico en las ciudades.

3.3.1. Modelado del tráfico de las ciudades

La información generada en tiempo real por sistemas *GPS* colocados en automóviles y dispositivos móviles facilita la medición y entendimiento del comportamiento del tráfico en las ciudades. Medir el tráfico permite crear modelos de predicción, detectar patrones de tráfico anormal y realizar estimaciones del promedio anual de tráfico diario o $AADT$ (del inglés *Average Annual Daily Traffic*). El $AADT$ es una medida ámpliamente utilizada en la planificación de la mejora de la infraestructura dedicada al transporte, es decir: calendarización del mantenimiento de caminos y calles, determinar qué nuevas autopistas se deben construir, *etc.*

El $AADT$ es la suma de todos los vehículos que pasan por un camino o calle durante un año dividido entre 365. En la actualidad existen aparatos especiales que han sido diseñados para realizar esta medición. No obtante es imposible colocarlos en todos los caminos o calles de una ciudad, por tal motivo los sistemas *GPS* ayudan a realizar estimaciones de este valor.

El análisis de la información proveniente de sistemas *GPS* es un reto ya que: i) existen una gran cantidad de dispositivos; ii) la información de todos estos dispositivos debe ser analizada casi en tiempo real; iii) todos los datos se deben almacenar para análisis futuros; iv) hay que lidiar con la información de dispositivos defectuosos o no sincronizados. El reto del punto ii se refiere al análisis que se realiza para modelar el tráfico existente al momento de realizar el análisis; mientras que el reto del punto iii se refiere a los análisis que permiten crear los modelos de predicción o bien estimar el $AADT$.

El modelado del tráfico de las ciudades se basa en el uso de análisis de agrupamiento particional [58]. Como se recordará el agrupamiento particional intenta dividir los datos en K grupos no aninados de objetos con características similares. En el modelado del tráfico se pueden utilizar medidas de similaridad tales como la densidad de automóviles al momento de realizar el análisis, vectores de conteos de automóviles, vectores de conteos por unidad de tiempo, *etc.*

Las herramientas de *Big Data* son de gran ayuda en el modelado del tráfico ya que

en los análisis de agrupamiento realizados para este fin no se conoce *a priori* el número de grupos que se está buscando [58]. Se realizan múltiples análisis de agrupamiento sobre los mismos datos utilizando diferentes número K de grupos. Posteriormente se determina cuál es el número K de grupos que brinda más información y en base ese resultado se realizan los análisis posteriores.

3.4. Resumen

Este capitulo presentó algunos problemas en los que se ha utilizado *Big Data*: la investigación de mercados, secuenciación del *DNA* y reconocimiento de patrones y modelado del tráfico.

La investigación de mercados (IM) tiene el propósito de estudiar a los clientes de una empresa para incrementar las ventas. Consiste de cuatro etapas: conocimiento de clientes, segmentación de clientes, creación de campañas publicitarias, y medición de la efectividad de las campañas publicitarias.

El conocimiento de clientes consiste en asignar atributos o etiquetas a cada cliente en base a información de: i) los tipos de productos que compra; ii) los comentarios o calificación que asigna a productos que ha comprado; iii) la calificación que da a comentarios de otros clientes en el sitio Web de la tienda o en las redes sociales como *twitter* y *facebook*. La segmentación de clientes es el proceso de identificar grupos de clientes considerando los productos que compran o algunas otras variables. Se realiza utilizando algoritmos de agrupamiento y clasificación [45]. Un tipo de campaña publicitaria que se puede crear utilizando *Big Data* son los sistemas de recomendación que muestran a los clientes productos que pueden ser de su interés, mediante la comparación los productos que compra con productos que compraron clientes similiares. La medición de la efectividad de campañas publicitarias consiste en determinar el incremento de ventas correspondiente. Esto se realiza por medio de algoritmos de regresión que relacionan ventas de productos (fecha, hora, lugar de aplicación, medio de comunicación, *etc.*) con una o más campañas publicitarias [46].

El *DNA* (del inglés *deoxyribonucleic acid*), es una molécula que guarda y transmite de generación en generación toda la información necesaria para el desarrollo de las funciones biológicas de un organismo. Para obtener una versión digital del *DNA* los científicos se apoyan de un proceso llamado secuenciación pero la enorme cantidad de datos generados contenidos en el *DNA* hace que las etapas que conforman este proceso sean lentas y costosas. *Big Data*, mediante el uso de cómputo en paralelo, logró disminuir considerablemente el costo y tiempo requerido.

El área geoespacial ha tenido retos muy grandes debido al gran volumen de datos generados por los satélites [57] y por los Sistemas de Posicionamiento Global o *GPS*. El modelado del tráfico de las ciudades es una de sus aplicaciones. Se basa en el uso de análisis de agrupamiento [58] sobre datos provenientes de sistemas *GPS*. Se realizan múltiples análisis de agrupamiento sobre los mismos datos utilizando diferentes número K de grupos. Posteriormente se determina cuál es el número K de grupos que brinda más información y en base ese resultado se realizan los análisis posteriores.

Capítulo 4

BDSP: *Big Data Start Platform*

Este capítulo presenta a *BDSP* (del inglés *Big Data Start Platform*), un sistema *Web* en el cual los usuarios pueden realizar análisis *Big Data* en cualquier momento y desde cualquier lugar utilizando sólo un dispositivo con conexión a Internet y un *browser*.

4.1. Motivación

En el capítulo 2 se describieron algunas herramientas y plataformas ampliamente utilizadas para realizar análisis de *Big Data*. Las herramientas y productos que se han presentado son muy poderosos pero requieren que el usuario las instale y configure adecuadamente antes de poder utilizarlas. En algunos casos realizar esta instalación puede ser un proceso complejo, en especial para usuarios que no pertenecen al área de computación y no tienen los conocimientos ni habilidades técnicas necesarias para realizar la instalación pero que tienen la necesidad de procesar y analizar grandes cantidades de datos.

Estas herramientas y productos requieren también que el usuario cuente con la infraestructura de *hardware* necesaria para almacenar sus datos, procesarlos y analizarlos. Por otra parte, el uso de los productos de *Big Data* puede conllevar el pago de una licencia de uso o suscripción.

Como se mencionó en el capítulo 2, algunos de los productos incluyen componentes opcionales con costo adicional que permiten procesar datos en paralelo utilizando herramientas como *Hadoop*, lo que permite el procesamiento de grandes cantidades de datos en un tiempo razonable. *Hadoop* y el *cluster* asociado deben ser instalados y configurados adecuadamente por el usuario antes de poder integrarlos a los productos de *Big Data*.

Otra opción es la siguiente. En los últimos años el cómputo en la nube (del inglés *cloud computing*) ha cambiado el paradigma de los modelos con el que se ofrecen servicios por Internet. Uno de estos modelos es el *Software* como Servicio o *SaaS* (del inglés *Software as a Service*) en donde la administración y mantenimiento del *hardware* y *software* es realizado por un tercero. Esto permite al usuario utilizar un *software* desde cualquier dispositivo con acceso a Internet compatible sin la necesidad de preocuparse en aspectos técnicos como la instalación o configuración. Algunos ejemplos de aplicaciones exitosas que han utilizado *SaaS* son: *Google Docs* [59] y *Adobe Creative Cloud* [60] y *Databricks* [15]. *Google Docs* permite almacenar, compartir y editar documentos de manera colaborativa. *Adobe Creative Cloud* es un conjunto de herramientas para la creación de gráficos y animaciones.

BDSP utiliza el modelo *SaaS* y por lo tanto los usuarios no tienen la necesidad de instalar ningún programa o de modificar la configuración de su equipo de cómputo — por supuesto, alguien debe instalar BDSP por primera vez pero esto debería ser tranparente para los usuarios finales. BDSP puede ser accedido/utilizado desde una computadora, desde una tableta, desde cualquier dispositivo inteligente con conexión a Internet que cuente con un navegador compatible con *HTML5*. Y es muy fácil de usar, e intuitiva para personas que saben que tipos de análisis son relevantes al problema que van a resolver. En el capítulo 5 se muestra la solución de varios problemas con BDSP. Básicamente solo se seleccionan opciones con el mouse y se especifican nombres de archivos (no hay necesidad de conocer que herramientas de análisis usar y como y con que parámetros.)

BDSP fue diseñado y construido para usar *Hadoop* por lo que es capaz de ejecutar

en paralelo una gran cantidad de algoritmos que se incluyen en el *framework Mahout* sin que el usuario final deba instalar ningún programa en su computadora o realizar alguna configuración en la misma. Además *BDSP* usa el sistema de archivos paralelo y distribuido *HDFS* de *Hadoop*, manera totalmente transparente al usuario, lo cual permite mejorar potencialmente el desempeño de análisis sobre grandes cantidades de datos.

Además de estas ventajas, *BDSP* es capaz de extraer datos desde una gran variedad de fuentes *Web* como *Twitter*, *Dropbox*, *Google Drive* entre otras. Es muy útil tener acceso inmediato y fácil a Twitter y Facebook por que son fuentes de información ampliamente utilizadas por muchas personas por que es información *real* sobre las opiniones de personas en todo el mundo sobre casi todos los temas del quehacer humano. Además, teniendo BDSP acceso a múltiples fuentes de datos, éstas son un ejemplo para extenderlo fácilmente para incluir otras fuentes que se requieran y así aumentar su integración e interoperabilidad con otros sistemas.

BDSP es libre y de código abierto por lo cual puede ser extendido para adaptarse a las necesidades específicas de cualquier usuario. El propósito de su diseño fue que sirviera de tres maneras: como prototipo inicial de proyectos *Big Data*, como plataforma base para extenderla según se requiera, y como vehículo de capacitación en análisis de datos y en desarrollo de *software Big Data*.

4.2. Arquitectura

La figura 4.1 muestra las capas principales de *BDSP*: la capa de interfaz gráfica de usuario, la capa de análisis de datos y la capa de manejo de datos.

La capa de interfaz gráfica o *GUI* (del inglés *Graphical User Interface*) está diseñada como un sitio *Web* que se ha implementado utilizando un servidor *Apache HTTP*, *MySQL*, *PHP*, *JavaScript*, *CSS3* y *HTML5*.

La capa de análisis de datos (AD) consta de servicios *Web* construidos con *PHP* que encapsulan frameworks de análisis y herramientas existentes. La AD proporciona

Figura 4.1: Capas de principales de *BDSP*.

algunos parámetros a estos *frameworks* y redirecciona los resultados generados a la *GUI* para que sean desplegados.

La capa de manejo de datos o (MD) consiste en servicios *Web* construidos con *PHP* y *MySQL*. Los servicios *Web* de la MD pueden acceder a archivos de datos locales y remotos. Los archivos remotos incluyen archivos en *Twitter*, *Facebook*, entre otros. La MD almacena los archivos obtenidos en *HDFS*, el sistema de archivos utilizado por *Hadoop*. La MD lleva un registro de los archivos almacenados en una tabla de base de datos *MySQL*, por lo que, de manera inmediata está información está disponible desde la *GUI* para que el usuario pueda realizar tareas de análisis sobre los datos.

En la figura 4.1, las flechas negras que apuntan a la AD y MD indican que una aplicación externa puede invocar directamente los servicios proporcionados por estas capas, adicionalmente a que el usuario puede utilizarlas mediante la *GUI*.

4.2.1. Estructura de base de datos

La figura 4.2 muestra las tablas *MySQL* utilizadas por *BDSP* y las relaciones existentes entre ellas. Estas tablas son: *usuario*, *configuracion*, *sesion*, *historial*, *proyecto*, *dato* y *analisis*.

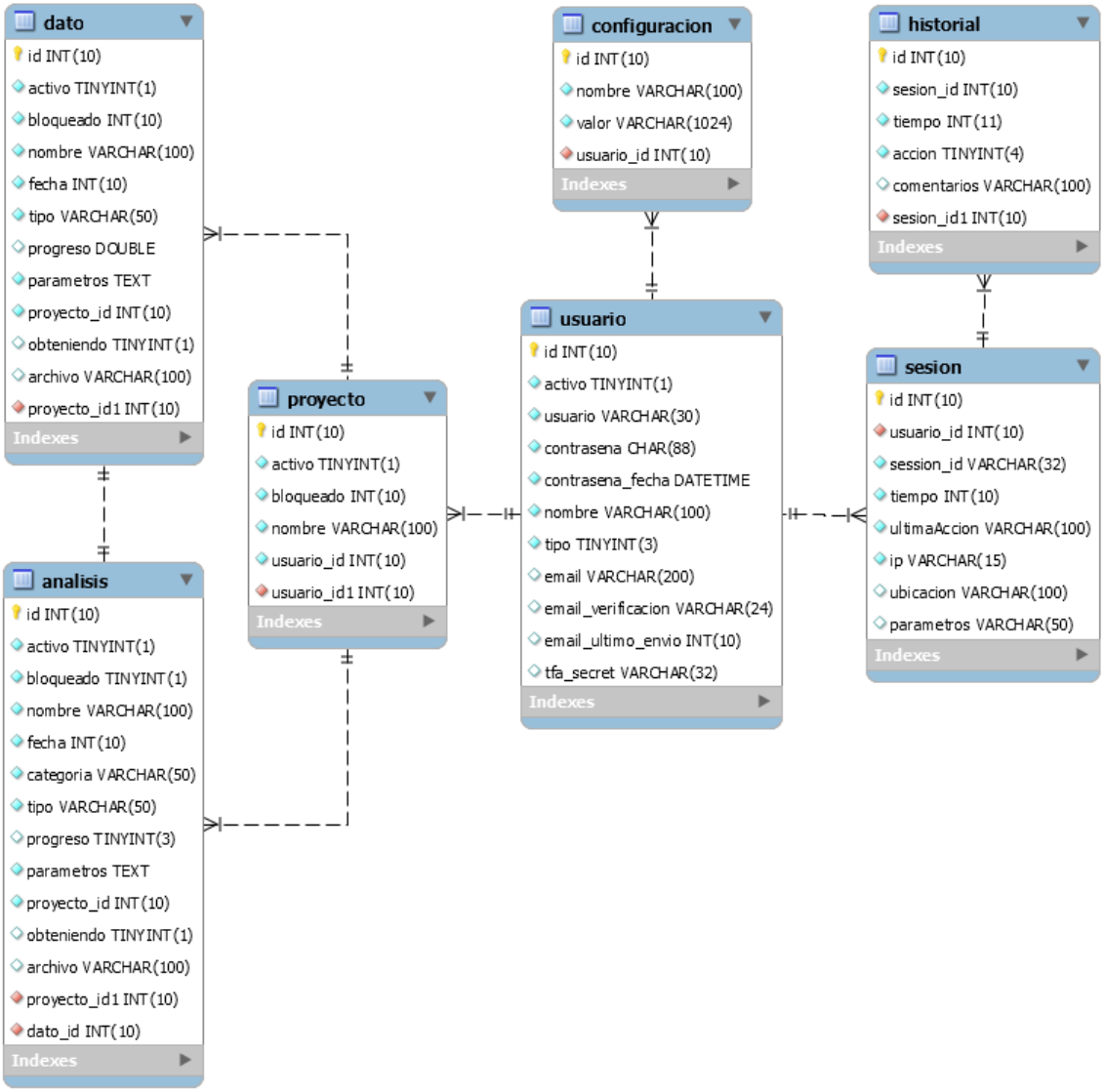

Figura 4.2: Estructura de la base de datos de BDSP.

La tabla usuario se encarga de almacenar todos los datos referentes al usuario, entre los que se incluyen su nombre de usuario y contraseña cifrada, la fecha del útlimo cambio de contraseña, su correo electrónico y el secreto utilizado para la autenticación de dos factores.

La tabla configuracion almacena todas las configuraciones realizadas por el usuario a *BDSP*. En la versión actual, su único uso es para almacenar las contraseñas de acceso

a otros servicios *Web* como *Dropbox* o *Google Drive*.

La tabla sesión almacena la información de cada inicio de sesión de los usuarios; entre los datos que se almacenan se encuentran la fecha y hora de la última actividad, descripción de la actividad realizada, dirección *IP* y ubicación aproximada. La tabla sesión es la que permite cerrar sesiones de manera remota. Cada vez que una acción es realiza por el usuario, *BDSP* verifica que la sesión siga activa. De no ser así, BDSP mostrará el formulario de inicio de sesión y cancelará cualquier acción realizada. Este funcionamiento se describe en la subsección 4.6.

La tabla historial lleva un registro de todas las acciones realizadas por el usuario.

La tabla proyecto almancena la información de los proyectos tal como su nombre y su estado.

Las tablas de datos y analisis almacenan los nodos de la cola de trabajo que se describen en la subsección 4.3. Entre los campos relevantes que almacena se encuentran el nombre como se identifica al dato o análisis en *BDSP*, el progreso de avance y los parámetros específicos de cada dato o análisis.

4.3. Capa de manejo de datos

La función principal de la capa de manejo de datos o MD es recuperar archivos y datos desde diferentes fuentes. Las fuentes de datos pueden ser remotas o locales. Las peticiones de recuperación pueden ser realizadas por el usuario mediante la *GUI* utilizando la opción *Datasets/Add Dataset*.

4.3.1. Manejo de peticiones de archivos

Cada petición de archivo es ingresada a una cola de trabajo de tipo *FIFO* (del inglés *First in, first out*) la cual es denominada *wq* y que es almacenada en una tabla de *MySQL*.

Un nodo en la *wq* corresponde a un registro en la tabla. Agregar un nodo a *wq*, es decir, realizar una operación *enqueue*, equivale a una operación *INSERT* de *SQL*.

Un nodo de la *wq* almacena el tipo de dato del archivo a obtener y los parámetros específicos que serán proporcionados al sitio de donde se recupere el archivo. Estos parámetros específicos se codifican y almacenan en la base de datos en formato *JSON* (del inglés *JavaScript Object Notation*), probablemente el formato de *marshalling* o serializacón con *JavaScript* más utilizado. Los nodos de la *wq* también almacenan el progreso de las solicitudes de recuperación de datos. Este progreso es usado por la *GUI* para desplegarlo al usuario como un porcentaje de avance.

El Proceso de Recuperación de Archivo o *FFP* (del inglés *File Fetch Process*) es un *script* desarrollado en *PHP*, el cual es responsable por realizar las peticiones de obtención de archivos de datos. Una operación *dequeue*, es decir, obtener el siguiente nodo de la lista (para llevar a cabo la transferencia de archivo correspondiente), no es propiamente una operación *dequeue*. En lugar de esto es una operación de invalidación del nodo relevante en la tabla de la *wq*. Los nodos en la *wq* nunca son eliminados y son mantenidos por motivos de historial y monitoreo. Sin embargo, siendo invalidados conceptualmente se encuentran eliminados (*dequeue*) mediante las siguientes dos operaciones *SQL*:

```
1  SELECT * FROM wq
2      WHERE   lock = 0 AND
3              progress < 100 AND
4              type = 'FF' LIMIT 1
5
6  UPDATE wq
7      SET     lock = 1
8      WHERE   id = 'N'
```

Donde *FF* especifica una operación de obtener un archivo y *N* corresponde al *id* del registro en la tabla de la *wq* que fue devuelto por la instrucción *SELECT*. Únicamente un resultado, o ninguno, es devuelto por la presencia de cláusula *LIMIT 1*.

Después de invocar estas dos operaciones *SQL*, si el *FFP* encuentra un nodo a procesar, entonces lo invalida estableciendo *lock=1*. Posteriormente, se determina el

tipo de archivo a obtener (*Twitter*, *Dropbox*, *etc.*) con el fin de redireccionar la petición a la *API* adecuada. La mayoría de las *APIs* tienen mecanismos de monitoreo, los cuales son utilizados por el *FFP* para realizar actualizaciones del campo progreso en el nodo (registro) correspondiente en la *wq*. Si la operación de recuperación se realiza de manera exitosa, *FFP* almacena el archivo en *BDSP* utilizando *HDFS*, y realiza la actualización *progreso=100*.

Si la operación de recuperación termina con errores, el *FFP* libera el nodo, actualiza *lock=0* y almacena el error ocurrido en el registro. Esta actualización hará que el nodo vuelva a ser elegible debido a que el progreso no será igual a 100 (ver la instrucción *SELECT*). *FFP*, volverá a internar obtener el archivo de dato en algún momento en el futuro. La mayoría de las *APIs* tienen funciones que permiten reiniciar la recuperación de datos desde el punto donde ocurrió algún error, lo cual es una optimización que proporciona mucha eficiencia, especialmente cuando se intentan recuperar archivos grandes. *BDSP* únicamente tiene implementada esa optimización para los archivos que son recuperados de *Twitter*.

Las dos operaciones *SQL* con las que se implementa la operación *"dequeue"* son seguras porque únicamente se ejecuta un proceso *FFP* que realiza la obtención de un archivo de datos a la vez, es decir, la recuperación de datos no se realiza en paralelo. Sería conveniente realizar la recuperación de archivos de manera concurrente con el fin de beneficiar a usuarios que necesitan recuperar múltiples archivos pequeños. Existen muchas maneras de realizar esta optimización, sin embargo este tema sale del alcance de esta tesis.

4.3.2. Archivos remotos

Las peticiones de archivos locales son procesados directamente por el servidor Apache *HTTP*. Las peticiones de archivos remotos las realiza BDSP mediante su capa MD. Entre los tipos de datos remotos que es posible recuperar mediante *BDSP* son *Twitter*, *Facebook*, *Dropbox* y *Google Drive*. A continuación describen cada uno de estos datos.

Twitter

Twitter[1] es un servicio de microblogging o publicaciones pequeñas. Es una red social que permite enviar mensajes de texto con un máximo de 140 caracteres, estos mensajes son denominados *tweets*. Los usuarios de *Twitter* pueden suscribirse para recibir los tweets publicados por otros usuarios, esta suscripción se llama *seguir a un usuario*. Por omisión los *tweets* son públicos pero es posible configurarlos para que sean privados y puedan ser visibles únicamente para los seguidores. Se ha implementado el acceso a Twitter porque es posible encontrar opiniones y comentarios de cualquier tema en todos los idiomas. Esta gran cantidad de comentarios puede ser de interés en muchas áreas. BDSP es capaz de acceder a los tweets públicos de todos los usuarios mediante la especificación de un término de búsqueda.

Facebook

Facebook[2] es la red social más popular del mundo con más de 1,650 millones de usuarios activos. *Facebook* proporciona un espacio a cada usuario llamado biografía. En su biografía los usuarios pueden publicar fotos, videos o comentarios. Estas publicaciones se conocen como estados. Los usuarios también pueden comentar sobre los estados de otros usuarios. Para poder ver el contenido de la biografía de otro usuario, es necesario que éste lo autorice por medio de una solicitud de amistad.

Es posible crear biografías públicas que pueden ser vistas sin necesidad de enviar solicitudes de amistad. Estas biografías especiales se conocen como páginas. Las empresas utilizan páginas para dar a conocer información sobre sus productos, servicios, promociones o cualquier información que consideren relevante. Todos los usuarios pueden ver y comentar los estados publicados en una página.

Se ha implementado el acceso a facebook porque es la red social más popular en la que se puede encontrar opiniones y comentarios de la mayoría de los temas en una gran cantidad de idiomas. Esta gran cantidad de comentarios puede ser de interés

[1]https://www.twitter.com/

[2]https://www.facebbok.com/

en muchas áreas. *BDSP* puede extraer estados y comentarios únicamente de páginas en facebook. No se ha contemplado el acceso a las biografías ya que por defecto esta información es privada.

Dropbox

Dropbox[3] es un servicio de alojamiento de archivos en la nube compatible con varios sistemas operativos como *Windows*, *Linux*, *Android*, entre otros. *Dropbox* permite a los usuarios almacenar archivos en línea y sincronizarlos entre distintas computadoras y dispositivos móviles. Sólo es necesario que se copie el archivo al directorio de *Dropbox* para que éste sea replicado en todos los dispositivos del usuario.

Se ha implementado el acceso a *Dropbox* porque es uno de los sistemas de almacenamiento en la nube más utilizado en el mundo. BDSP puede acceder a todos los archivos almacedos por el usuario.

Google Drive

Google Drive[4] es un servicio de alojamiento de archivos en la nube compatible con varios sistemas operativos como *Windows*, *Linux*, *Android*, entre muchos otros. *Google Drive* permite a los usuarios almacenar archivos en línea y sincronizarlos entre distintas computadoras y dispositivos móviles.

Se ha implementado el acceso a *Google Drive* porque ofrece hasta 15GB de almacenamiento gratuito, siendo el servicio que ofrece la mayor capacidad de almacenamiento sin costo. BDSP puede acceder a todos los archivos almacedos por el usuario.

4.3.3. Módulos funcionales y organización de archivos

La capa de manejo de datos (MD) está conformada por los módulos mostrados en la figura 4.3. El módulo *MDmain*, en la parte superior de la figura, es el encargado de obtener y procesar las peticiones de archivos locales y remotos que el usuario

[3]https://www.dropbox.com/

[4]https://drive.google.com/

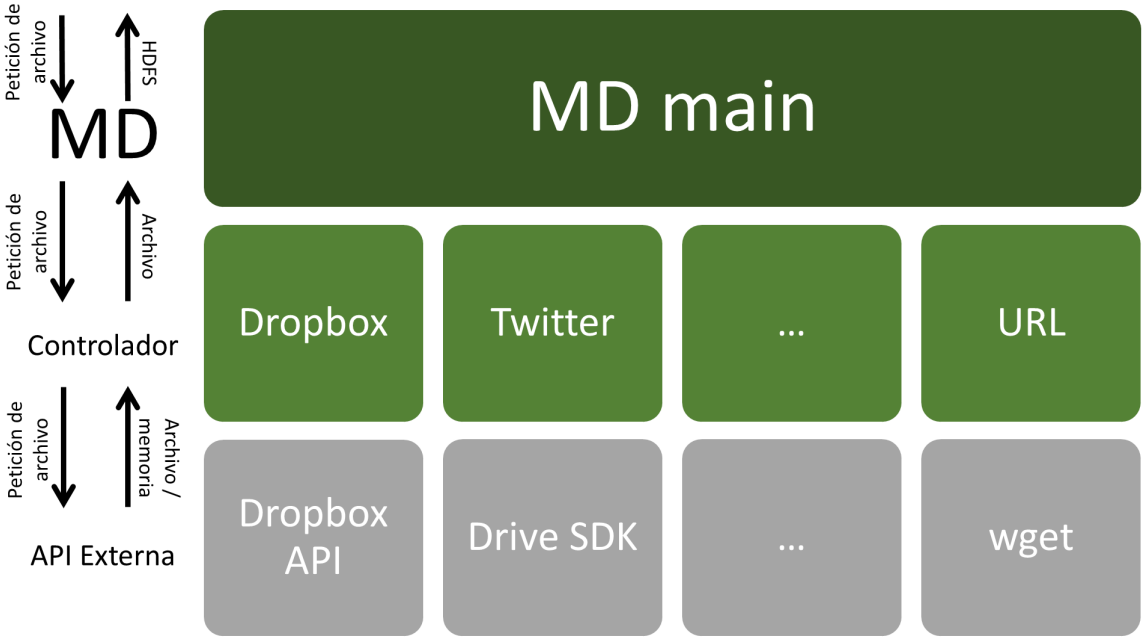

Figura 4.3: Módulos de la capa de manejo de datos.

especifica por medio de la *GUI*. Cada petición corresponde a un nodo de la cola de trabajo *FIFO wq*, descrita en la sección 4.3.1, y la cual es manejada como una tabla en una base de datos *MySQL*.

Procesamiento periódico de peticiones de archivos

El modulo *MDmain* verifica periódicamente la *wq* para comprobar si hay alguna petición de archivos locales o remotos. *MDmain* es un proceso que es creado por el proceso *CRON* de *Linux* [61] como se describe en el siguientes párrafo. *MDmain* obtiene una sola petición de archivos de la cola *wq* y la procesa como se detalla más adelante y termina. Si no hay peticiones *MDmain* termina inmediatamente.

CRON es un demonio de *Unix/Linux* que permite ejecutar automáticamente comandos o *scripts* en una hora o fecha específica. Es usado normalmente para realizar tareas administrativas tales como respaldos. Los procesos que *CRON* ejecutará automáticamente se especifican como una línea de comandos dentro de alguno de los

archivos de configuración que se encuentran en `/etc/cron.d/`. Cada una de estas líneas consta de 6 elementos que se muestran en la figura 4.4. Los cinco primeros parámetros corresponden, respectivamente, al minuto, hora, día del mes, mes y días de la semana en los que el comando especificado con el sexto parámetro se debe ejecutar [61]. Si alguno de los primeros cinco parámetros se configura con un asterísco (*) significa que se ejecutará durante todos los elementos que son representados por el parámetro.

CRON crea instancias del proceso *MDmain* a partir del código fuente en el archivo:

`__BDSP__/capa-md/md.php`

donde `__BDSP__` se refiere al directorio donde se instaló *BDSP*. En el caso de un servidor *Linux Debian*, `__BDSP__` corresponde al directorio `/var/www/BDSP/`, tal y como se indica en el apéndice B. Los procesos *MDmain* se ejecutan cada 15 segundos en la versión actual, con los siguientes comandos que se especifican en el archivo `/etc/cron.d/bdsp`

```
1  * * * * * php __BDSP__/capa-md/md.php
2  * * * * * (sleep 15 ; php __BDSP__/capa-md/md.php)
3  * * * * * (sleep 30 ; php __BDSP__/capa-md/md.php)
4  * * * * * (sleep 45 ; php __BDSP__/capa-md/md.php)
```

Como se mencionó anteriormente, los asteríscos indican que el comando se ejecutará durante todos los elementos que son representados por el parámetro, es decir, estos comandos se ejecutarán todos los minutos de todas las horas de todos los días de todos los meses. Esto significa que los comandos se ejecutarán cada minuto. Nótese que los comandos de las líneas 2, 3 y 4 inician con un `sleep`, por lo que, aunque el proceso se crea cada minuto, se quedará dormido por el número de segundos especificados. En la práctica, *MDmain* se ejecutará cada 15 segundos.

Transferencias de archivos

En cada petición de archivo que es obtenida por *MDmain* de la cola *wq* se especifica: i) la *fuente del archivo* (local, *Dropbox*, *Google Drive*, *etc.*); ii) un bloque de

```
.---------------- minuto (0 - 59)
|  .------------- hora (0 - 23)
|  |  .---------- día del mes (1 - 31)
|  |  |  .------- mes (1 - 12)
|  |  |  |  .---- día de la semana (0 - 6)
|  |  |  |  |
*  *  *  *  *  comando para ser ejecutado
```

Figura 4.4: Configuración del comando *CRONTAB*.

parametros especificos requeridos por la *API* a utilizar para obtener el archivo (ver apartado **Configuraciones** abajo); y iii) el identificador del usario que está solicitando el archivo.

La *fuente del archivo* es una cadena que *MDmain* utiliza para identificar, cargar, y ejecutar el procedimiento encargado de preparar la solicitud del archivo a la *API* externa. Este procedimiento, en *BDSP*, es llamado *controlador*. Hay un *controlador* para cada fuente de archivos y algunos de éstos se observan en la parte central de la figura 4.3. El código fuente de los todos los controladores encuentra en:

`__BDSP__/capa-md/fuentes/{fuente del archivo}.php`

La función de un controlador es preparar la solicitud de un archivo para obtenerlo mediante el uso del *API* de la fuente correspondiente, proporcionada por los servicios externos a los que *BDSP* puede acceder. Esta preparación consiste en procesar el bloque de parámetros específicos de cada *API*, que están codificados en formato *JSON*, y convertirlos a un conjunto de llamadas a funciones de la *API* del servicio externo. Las llamadas a funciones realizadas depende de la *API*. Por ejemplo, la petición de un archivo en *Dropbox* conlleva los siguiente pasos: 1) autenticar a *BDSP* ante *Dropbox*; 2) autenticar al usuario propietario del archivo a transferir ante *Dropbox*; 3) especificar la ruta del archivo a obtener; y 4) solicitar a *Dropbox* que empiece la transferencia del archivo. Se muestran las diferencias con otras *APIs* o solicitudes en el apartado **Configuraciones**.

Una vez que se solicita la transferencia de un archivo a la *API* externa, *MDmain* espera, dentro del procedimiento del controlador que solicita la transferencia, a que termine la tansferencia. Si ocurre un error, el controlador se hace cargo de manejarlo

si es posible. De lo contrario guarda un informe de error que incluye los parámetros proporcionados a la *API*. Este informe estará disponible únicamente para el administrador de *BDSP*. Por ejemplo, un error que puede ser manejado por el controlador es un problema de conexión a Internet, si esto ocurre se reintentará inmediatamente obtener el archivo, en caso de volver a ocurrir un problema de conexión el controlador notificará el error y liberará el nodo para intentarlo de nuevo después.

Si la transferencia de un archivo es exitosa, una copia del mismo será almacenada en el sistema de archivos local del servidor *Web* donde se está ejecutando *BDSP*. *MDmain* procede entonces a hacer una copia del archivo al sistema de archivos *HDFS* de *BDSP*, terminando el procesamiento. Nótese que no todas las APIs retornan archivos, por lo que es responsabilidad del controlador guardar el resultado en un archivo. Por ejemplo, *Twitter* retorna los *tweets* obtenidos en un buffer que se queda en memoria, y el controlador de *BDSP* para *Twitter* es el encargado de guardarlos en un archivo local y devolver a *MDmain* su nombre. *MDmain* procede entonces a hacer una copia del archivo al sistema de archivos *HDFS* de *BDSP*.

Configuraciones

Para que *BDSP* pueda hacer uso de las *APIs* de *Dropbox*, *Google Drive*, *Twitter* y *Facebook*, es necesario registrar a *BDSP* en la consola de desarrolladores de cada uno de estos servicios; esto ya se hizo. Este registro genera un par de llaves que sirven para autenticar a *BDSP* ante estos servicios. Nos referiremos a este par de llaves como *llaves de autenticación de BDSP*.

Para que *BDSP* pueda acceder los archivos en *Dropbox*, *Google Drive*, etc. de un usuario en particular, tal usuario debe tener una cuenta de, y autenticarse una única vez (con su *nombre de usuario* y su *password*) en, tales servicios. Cuando el usuario realiza tal autenticación, BDSP recibe un par de llaves de autenticación que son usadas en subsecuentes accesos a archivos sin la necesidad de que el usuario se vuelva a autenticar. Es posible que la primera autenticación de usuario ocurra sin que el usuario lo haga a través de BDSP. Por ejemplo, si el usuario entra (se autentica) primero a

su cuenta de Dropbox y luego entra a BDSP y solicita un archivo de su cuenta de Dropbox, es muy probable que Dropbox ya no requiera una segunda autenticación del usuario. Esto es porque, gracias al uso de *cookies* los cuales almacenan información sobre la sesión de un usuario en un browser, las sesiones de Dropbox (como las de muchos otros sistemas web) se mantienen abiertas y, al Dropbox solicitar la segunda autenticación el browser responderá con la información en la cookie y se abrira una nueva sesión sin la necesidad de autenticación por parte del usuario.

BDSP maneja dos usuarios *dummy*, uno para Twitter y otro para Facebook, y sus llaves de autenticación de usuario correspondientes para poder acceder a información pública de muchos usuaros en estos dos servicios. (La información de un solo usuario en tales servicios no parece relevante y BDSP no maneja esta opción.) Con el usuario dummy de cada servicio, cualquier usuario de BDSP puede acceder a información pública sin tener que ligar su cuenta de Twitter or Facebook. La creación de los usuarios dummy es necesario pues, como se muestra adelante, la API requiere la especificación de las llaves de autenticación de un usuario. Estas llaves se encuentran almacenadas en el archivo `__BDSP__/fdw_configuracion.php`

Para acceso a archivos en Dropbox y Google Drive, BDSP si maneja llaves de autenticación por cada usuario, y para obtenerlas cada usuario debe dar su autorización explícita de acceso a su cuenta una vez que sea ha autenticado explícitamente (o implícitamente como descrito antes) ante tales servicios. Esta autorización la da un usuario mediante una página de autorización del sitio *Web* del servicio que se pretende acceder, ver figura 4.5 (nótese que la *URL* de la página de autorización corresponde al sitio *Web* de *Dropbox*.). Para que esta página de autorización se despliegue al usuario, *BDSP* redirige el navegador del usuario a una *URL*, generada por la *API* externa del servicio correspondiente (ver figura 4.3), y que está asociada a las *llaves de autenticación de BDSP*. El usuario deberá, dentro de esta página dar click en el botón de autorización, después la *API* externa proporcionará a *BDSP* el par de llaves de autenticación que identificarán al usuario en subsecuentes accesos a archivos. *BDSP* no almacena, ni tiene acceso en ningún momento a la contraseña utilizada por

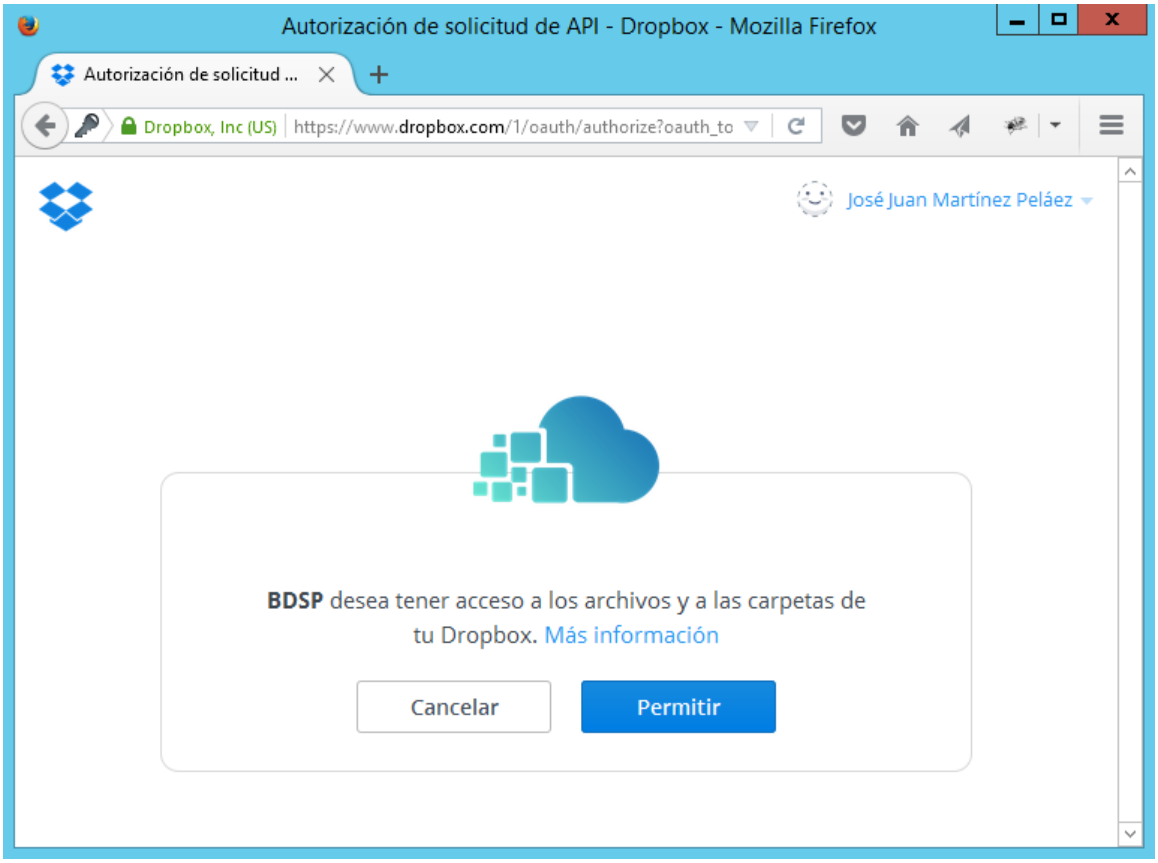

Figura 4.5: Página que autoriza a BDSP acceder a los datos de un usuario en *Dropbox*.

usuario para acceder a servicios externos.

Una vez que se tienen las llaves de autenticación de BDSP y de cada usuario (real o dummy), el acceso a cada servicio o fuente de datos a través de su API involucra los siguientes pasos:

Dropbox

1. Se proporcionan las *llaves de autenticación de BDSP*.
2. Se proporcionan las *llaves de autenticación de usuario*.
3. Se especifica la ruta del archivo a transferir.
4. Se solicita iniciar la transferencia.

Google Drive

1. Se proporcionan las *llaves de autenticación de BDSP*.

2. Se proporcionan las *llaves de autenticación de usuario*.

3. Se especifica el ID del archivo a transferir.

4. Se solicita iniciar la transferencia.

Twitter

1. Se proporcionan las *llaves de autenticación de BDSP*.

2. Se proporcionan las *llaves de autenticación de usuario*.

3. Se especifica el término de búsqueda.

4. Se especifica la cantidad de *tweets* a obtener, el idioma y/o el rango de fechas a utilizar.

5. Se solicita iniciar la recuperación de *tweets*.

Facebook

1. Se proporcionan las *llaves de autenticación de BDSP*.

2. Se proporcionan las *llaves de autenticación de usuario*.

3. Se especifica la página de búsqueda o el identificador del estado.

4. Se especifica la cantidad de comentarios a obtener.

5. Se solicita iniciar la recuperación de comentarios.

MySQL

1. Se proporciona el *host* donde se encuentra el servidor *MySQL*.

2. Se especifica el nombre de usuario y contraseña.

3. Se especifica el nombre de la base de datos.

4. Se solicita iniciar la recuperación de datos utilizando la siguiente consulta:

```
SELECT * FROM {tabla} LIMIT {max}
```

donde tabla es el nombre de la tabla de la que se extraerán los datos y max es el número máximo de registros a obtener.

4.4. Capa de análisis de datos

La función principal de la Capa de Análisis de Datos o AD es realizar diferentes tipos de análisis sobre los archivos de datos que se encuentran disponibles en el *HDFS*

Figura 4.6: Capa de Análisis de Datos de *BDSP*.

de *BDSP*. Las tareas de análisis son especificadas por el usuario a través de la *GUI* en la opción *Analyses/Add analysis*.

4.4.1. Manejo de peticiones de análisis

Cada una de las peticiones es agregada a la misma cola de trabajo (*wq*) usada por la capa de manejo de datos (MD). Por este motivo, el procesamiento de las tareas de análisis es muy similar con respecto a las operaciones de *enqueue* y *dequeue* en la misma tabla de la *wq*. Sólo hay una excepción: las tareas de análisis de datos son identificadas en la tabla *wq* con 'DA' en lugar de 'FF' y un solo Proceso de Análisis de Datos o *DAP* (del inglés *Data Analisys Process*), un script en *PHP*, es responsable por el procesamiento de las peticiones de análisis de datos.

Tal y como ocurre con el *FFP* (*File Fetch Process*, sección 4.3.1), el proceso *DAP* realiza los análisis de datos de manera serializada y puede ser optimizada de varias maneras. *BDSP* actualmente utilizada sólo un cluster *Hadoop* que es asignado de manera total a cada una de los procesos de análisis. Si se particiona el *cluster*, múltiples tareas de análisis podrían realizarse de manera simultánea con lo que se mejoraría el *throughput*. Otra alternativa es que los usuarios puedan configurar en su cuenta de *BDSP* acceso a un *cluster* privado *Hadoop* gestionado por ellos, es decir, un *cluster* adicional al proporcionado por *BDSP*, permitiendo con esto la distribución de manera concurrente de los tareas de análisis. Éstas son algunas de las optimizaciones que pueden ser realizadas como trabajo futuro.

La figura 4.6 muestra la organización de la AD. Esta capa encapsula *frameworks* y herramientas existentes pero puede ser extendida para agregar una funcionalidad específica como el análisis de sentimiento que se describe a continuación. Los *frameworks* que actualmente están integrados son: el *Toolkit* de Lenguaje Natural o *NLTK* (del inglés, *Natural Language Toolkit*); la Arquitectura General para la Ingeniería de Texto o *GATE* (del inglés, *General Architecture for Text Engineering*) y *Apache Mahout*. Estos *frameworks* se ejecutan sobre *Hadoop* por defecto pero pueden ser ejecutados en una sola computadora de manera local.

Cuando el proceso *DAP* realiza la operación *"dequeue"* al primer nodo de la *wq*, primero identifica el tipo de análisis involucrado e invoca al *framework* adecuado a través de la llamada a la API correspondiente para realizar el análisis. Si el análisis es exitoso, el resultado es almacenado como un archivo en el *HDFS* de *BDSP*; de lo contrario una notificación de error es mostrada al usuario.

4.4.2. Análisis de sentimiento

En la versión actual de *BDSP*, sólo *NLTK* es utilizado para la realización del análisis de sentimiento, a continuación se describe el procedimiento. Primero, y con el fin de mejorar el resultado del análisis, los datos de entrada son preprocesados mediante el uso de expresiones regulares, escritas en *PHP*, según las recomendaciones realizadas por Smailovic *et.al.* [62]:

- Remplazar los nombres de usuarios como *@UsuarioDeTwitter* por la palabra *Usuario*, esto con el fin de reducir el tamaño del diccionario utilizado (bolsa de palabras).
- Remplazar los enlaces *Web* por la palabra *URL*, por la misma razón.
- Remover letras repetidas con más de dos ocurrencias como en *Holaaaaaa*, debido a que estas palabras no se encuentran en los diccionarios.
- Remplazar los signos de exclamación y de interrogación, los cuales expresan

cierto tipo de emoción dependiendo del contexto por las palabras *Exclamación* y *Pregunta*, por la misma razón.

Después se ejecuta *NLTK* con los datos preprocesados como entrada para realizar una separación de *tokens*. Se utiliza el diccionario llamado *SentiWordNet* para obtener un puntaje para cada *token*. Finalmente el proceso *DAP* realiza el cómputo de todos los puntajes individuales obtenidos del término que se está analizando. Por ejemplo *Big Data*.

El preprocesamiento utilizado para análisis de sentimiento que se ha descrito puede ser añadido para otros tipos de análisis o archivos de datos. Por ejemplo, Procesos de Extracción, Transformación y Carga, conocidos como *ETL* (del inglés, *Extract, Transform and Load*) se pueden añadir para lidiar con problemas como valores faltantes o fuera de rango.

4.4.3. Otros análisis

La biblioteca *Apache Mahout* es utilizada para ejecutar los demás tipos de análisis que se encuentran disponibles en *BDSP*: regresión (simple, múltiple, lineal y no lineal), clasificación (*C4.5*, *ID3*), agrupamiento (*K-Means*, *Fuzzy C-Means*), muestreo (Gibbs, Metropolis-Hastings, Monte Carlo).

Mahout se compone de implementaciones libres de algoritmos de minería de datos, aprendizaje de máquina y filtros colaborativos [50]. La mayoría de los algoritmos implementaods en *Mahout* pueden ser ejecutados sobre *Hadoop*; algunos algoritmos pueden ser ejecutados sólo en entornos de una sola computadora (sin paralelizar), mientras que otros pueden ser ejecutados en *Spark*, una plataforma similar a *Hadoop* diseñada para el cómputo de datos en memoria que se ejecuta mucho más rápido que *Hadoop*.

4.4.4. Archivos de resultados

Un archivo de resultados (de un análisis) es generado cada vez que se finaliza un análisis sobre un conjunto de datos. Este archivo de resultados contiene la salida del análisis especificado por el usuario. *BDSP* almacena los archivos de resultados en *HDFS*.

Nótese que no es posible especificar un análisis sobre el vacío, es decir, un análisis se puede especificar solamente como una operación sobre un archivo de datos existente. Al especicar un análisis, se debe proporcionar el nombre del archivo de resultados, el tipo de análisis a realizar, seleccionar el archivo de datos a analizar y seleccionar el proyecto al que lo agregarán.

4.4.5. Módulos funcionales y organización de archivos

Los módulos que componen la capa de análisis de datos (AD) se observan en la figura 4.7. La interacción entre estos módulos es similar a la interacción existente entre los módulos de la capa de manejo de datos: un modulo principal comanda un análisis por medio de invocar procedimientos en ambos un modulo controlador y una API externa. El modulo principla de la capa AD es llamado ADmain y también se ejecuta periodicamente cada 15 segundos gracias al proceso CRON de Linux con los comandos que se especifican en el archivo `/etc/cron.d/bdsp` (ver detalles en la sección 4.3.3).

La función de *ADmain* es obtener el siguiente nodo de la pila *wq*, descrita en la sección 4.3.1. Si es posible obtener un nodo de la pila, *ADmain* lo procesará y posteriormente morirá. En caso de no obtener ningún nodo, *ADmain* morirá inmediatamente. Cada nodo contiene la especificación de un análisis, programado por el usuario mediante la *GUI*, e incluye: 1) el tipo de análisis a realizar; 2) el archivo de datos a procesar con tal análisis; y 3) los parámetros específicos requeridos por el análisis representados en formato JSON.

ADmain utiliza el tipo del análisis para cargar el procedimiento necesario pa-

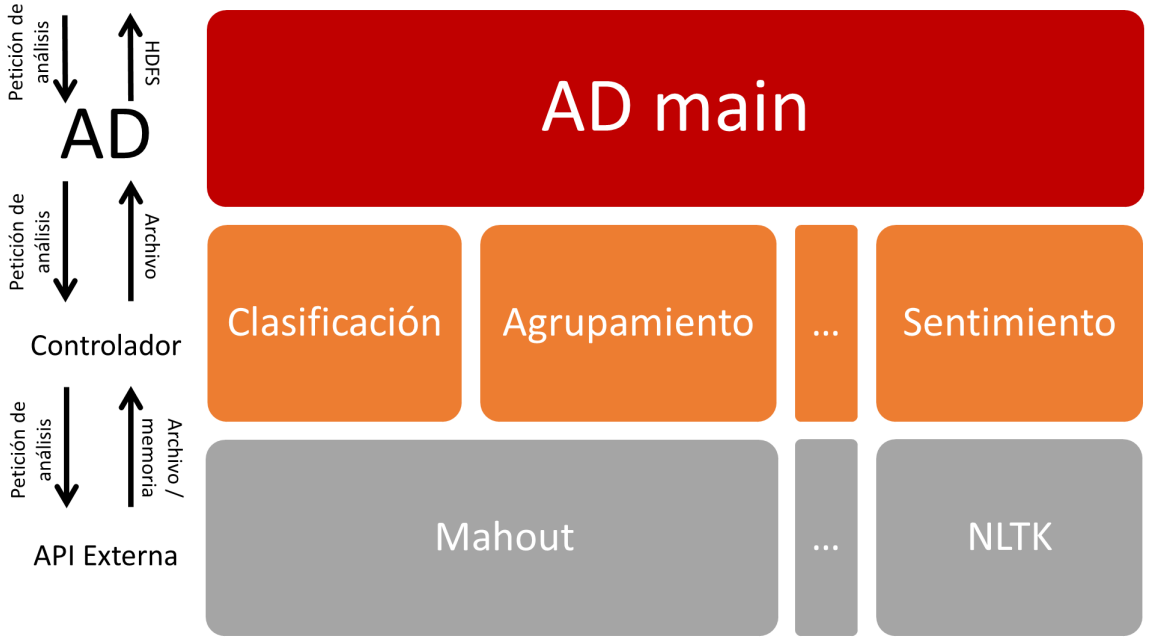

Figura 4.7: Módulos de la capa de análisis de datos.

ra realizar el análisis. Este procedimiento, en *BDSP*, se denomina controlador. Los archivos de los controladores se encuentran en:

`__BDSP__/capa-ad/analisis/{tipo de análisis}.php`

ADmain proporciona al controlador la ruta en *HDFS* del archivo de datos a utilizar y los parámetros específicos requeridos por el análisis en forma de arreglo de *PHP*. El controlador invoca entonces a la API del análsis correspondiente. En nuestra implementación, la API corresponde a un script del shell de Linux que invoca el paquete de análisis relevante pasándole los parámetros que se hayan especificado, como se muestra en el código a continuación:

```
<?php
function RealizarAnalisis($archivo_hdfs, $parametros) {
// comandos de preparación de parámetros

// "<pre>"; y  "</pre>"; indican al Browser que lo que sigue (la
```

```
// salida generada por el análisis a ejecutar) es texto plano

 echo "<pre>";
 //                          API del análsis a realizar
 echo exec("sudo /__BDSP__/capa-ad/analisis/{tipo de analisis}.sh ...
                 $archivo_hdfs param1 param2 ...");
 echo "</pre>";
}
?>
```

Nótese que los resultados de un análisis corresponden a la salida de la ejecución de *scripts* del shell de Linux como se comenta en el código arriba. Cada script de Linux corresponde a la invocación manual, con la línea de comandos, de la herramienta de análisis (Mahout, etc.) relevante incluyendo sus parámetros, etc.

En la versión actual de BDSP, los controladores de AD son incapaces de manejar errores. En caso de ocurrir un error, éste se incluirá como parte del archivo de resultados — la salida generada por el análisis. Por ejemplo, si un usuario intenta realizar un análisis de regresión sobre campos que no son numéricos, el análisis fallará. La salida generada por la biblioteca encargada de realizar el análisis se incluirá tal cual en el archivo de resultados permitiendo al usuario ver el porqué del fallo.

Si el análisis se realiza de manera exitosa, o bien si ocurre algún fallo se generará un archivo de resultados, el cual será almacenado de manera local en el servidor *Web* en donde se ejecuta *BDSP*. Este archivo será proporcionado a *ADmain* para que lo almacene en HDFS.

4.5. Capa de Interfaz Gráfica

La capa de interfaz gráfica o *GUI*, corresponde a la interfaz del sitio *Web* donde los usuarios pueden manejar proyectos, archivos de datos y archivos de análisis de manera sencilla. La interfaz gráfica se programó con *PHP* del lado del servidor y se

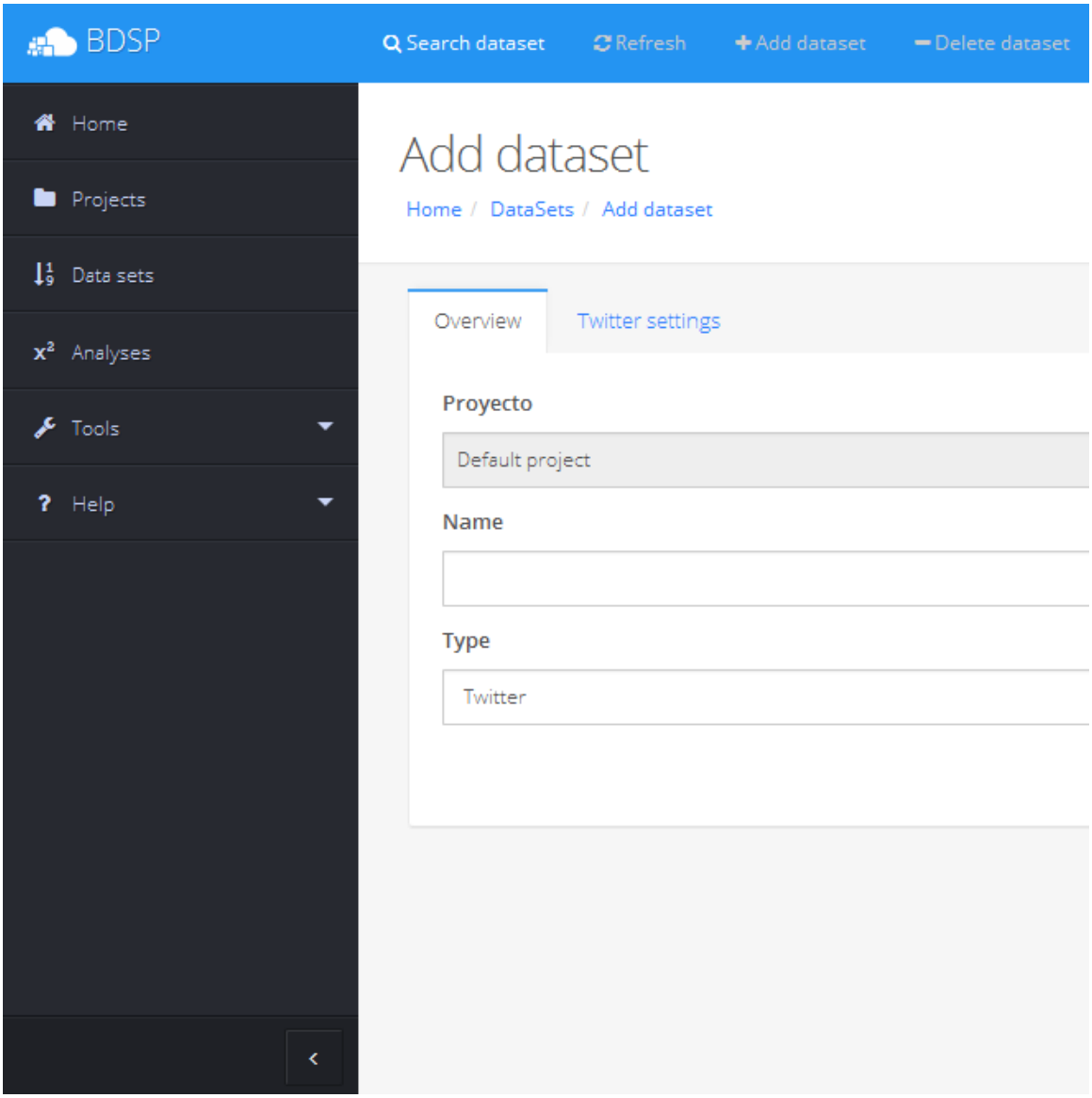

Figura 4.8: Parte izquierda de la vista de la interfaz gráfica de *BDSP* en una computadora de escritorio.

utilizó *HTML5*, *CSS3* y *JavaScript* del lado del cliente en donde se utilizaron las bibliotecas *jQuery* y *Bootstrap*.

jQuery[5] es una biblioteca libre y de código abierto que simplifica la modificación de documentos *HTML* utilizando *JavaScript*. Permite manipular de manera senci-

[5]https://www.jquery.org

lla el árbol *DOM* (del inglés *Document Object Model*), manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX* (del inglés *Asynchronous JavaScript And XML*). Esta biblioteca fue creada por John Resig y presentada el 14 de enero de 2006 en el *BarCamp NYC*. En la actualidad *jQuery* es la biblioteca de *JavaScript* más utilizada.

Twitter Bootstrap[6], mejor conocido únicamente como *Bootstrap* es un *framework* libre para diseño de sitios y aplicaciones *Web*. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en *HTML* y *CSS*, así como algunas extensiones de *JavaScript*.

4.5.1. Manejo de proyectos

El manejo de proyectos y archivos incluye las siguientes tareas: agregar, modificar, eliminar y visualizar. Al ser la interfaz de un sistema *Web* es posible realizar este manejo desde cualquier dispositivo con conexión a Internet utilizando un navegador.

Un proyecto es un contenedor en donde es posible colocar archivos de datos y especificar análisis sobre archivos. Los proyectos se impementaron con el fin de permitir al usuario organizar sus archivos de datos y análisis relacionados. Todos los archivos de datos y análisis realizados se deben colocar dentro de un proyecto o de lo contrario *BDSP* lo colocará en el *Default Project* o proyecto por defecto.

Para cada archivo de datos, los usuarios deben proporcionar un nombre con el que se identificará al archivo dentro de *BDSP*, su tipo u origen de donde *BDSP* debe obtenerlo y especificar el proyecto al que desean agregarlo.

4.5.2. Diseño *Web* Adaptable

Una de las principales características de la interfaz *Web* desarrollada es que cuenta con un Diseño *Web* Adaptable (en inglés *Responsive Design*) que le permite modificar su apariencia según las características del dispositivo que se esté utilizando para vi-

[6]http://www.getbootstrap.com/

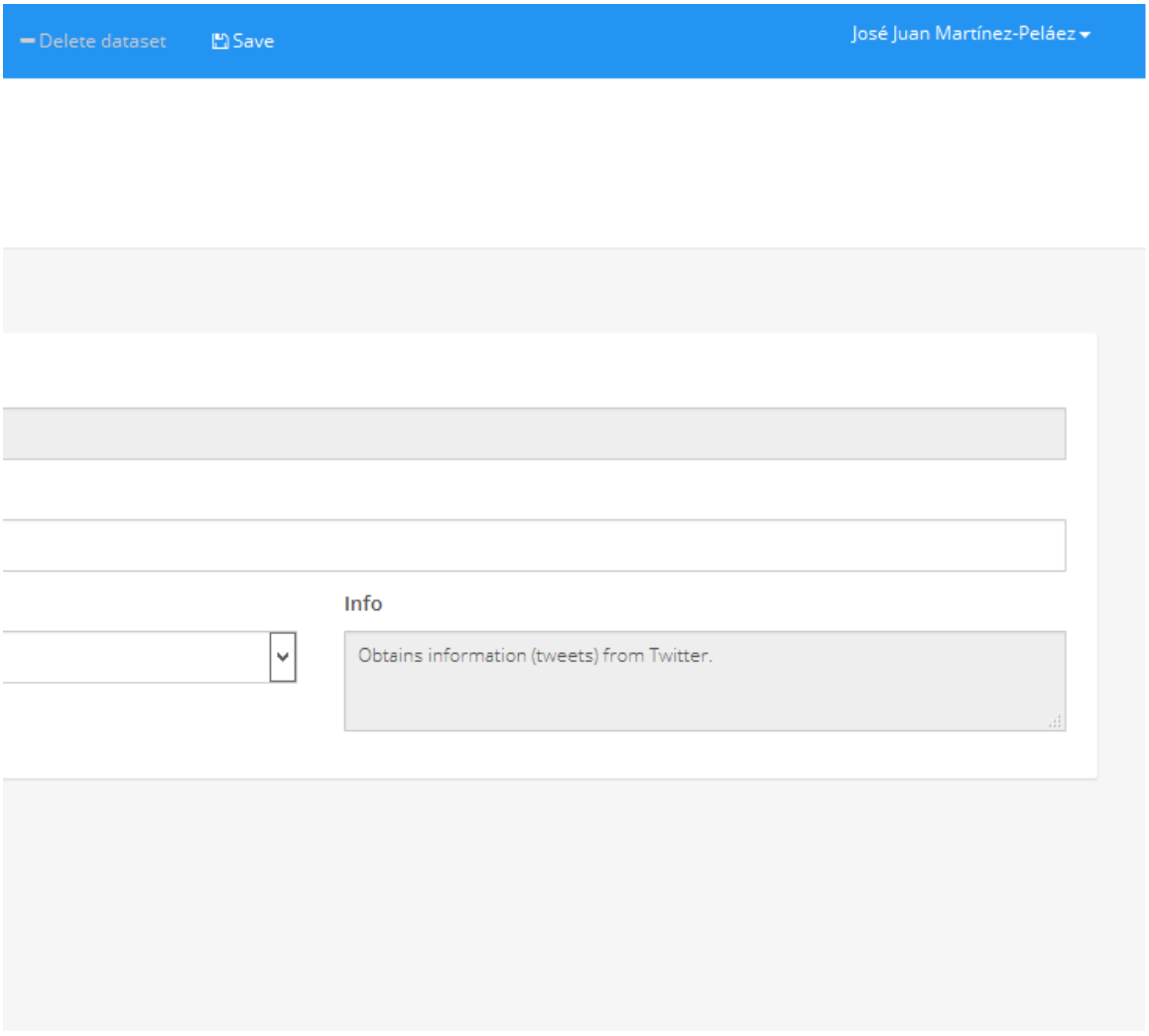

Figura 4.9: Parte derecha de la vista de la interfaz gráfica de *BDSP* en una computadora de escritorio.

sualizarla. Las figuras 4.9 y 4.10 muestran el sistema visualizado en una computadora de escritorio y en un teléfono. En estas figuras lo que se adaptó fue el desplegado del menú. En la figura 4.10, se redujo el menú a un ícono con líneas horizontales en la parte superior de la pantalla. Al dar click en este ícono se despliega el menú, tal y como se muestra en la figura 4.11.

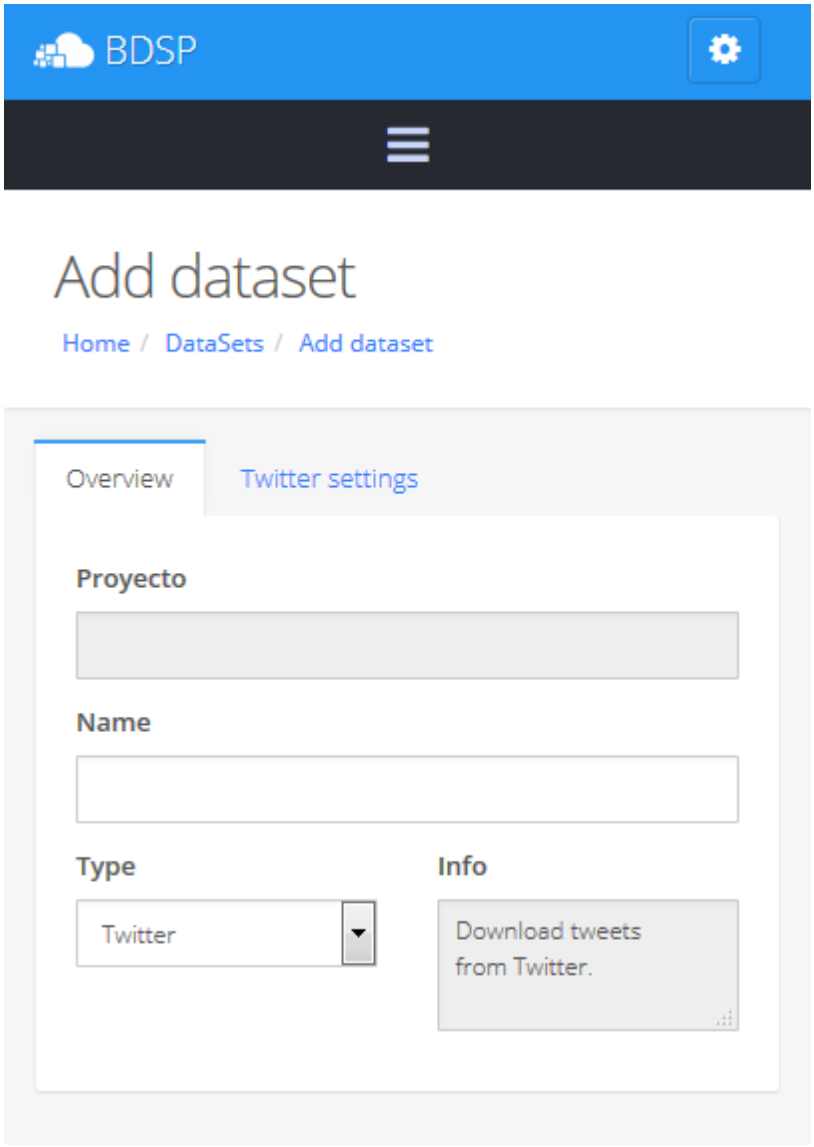

Figura 4.10: Vista de la interfaz gráfica de *BDSP* en un *smartphone*.

4.5.3. Registro de usuarios

La interfaz permite el registro de nuevos usuarios utilizando un formulario donde se debe proporcionar un nombre de usuario, contraseña y un correo electrónico. Se enviará un correo electrónico para confirmar que la dirección proporcionada por el usuario es correcta.

Una vez que se ha registrado un usuario éste puede modificar su perfil. La interfaz permite cambiar la contraseña, cambiar el correo electrónico y habilitar o deshabili-

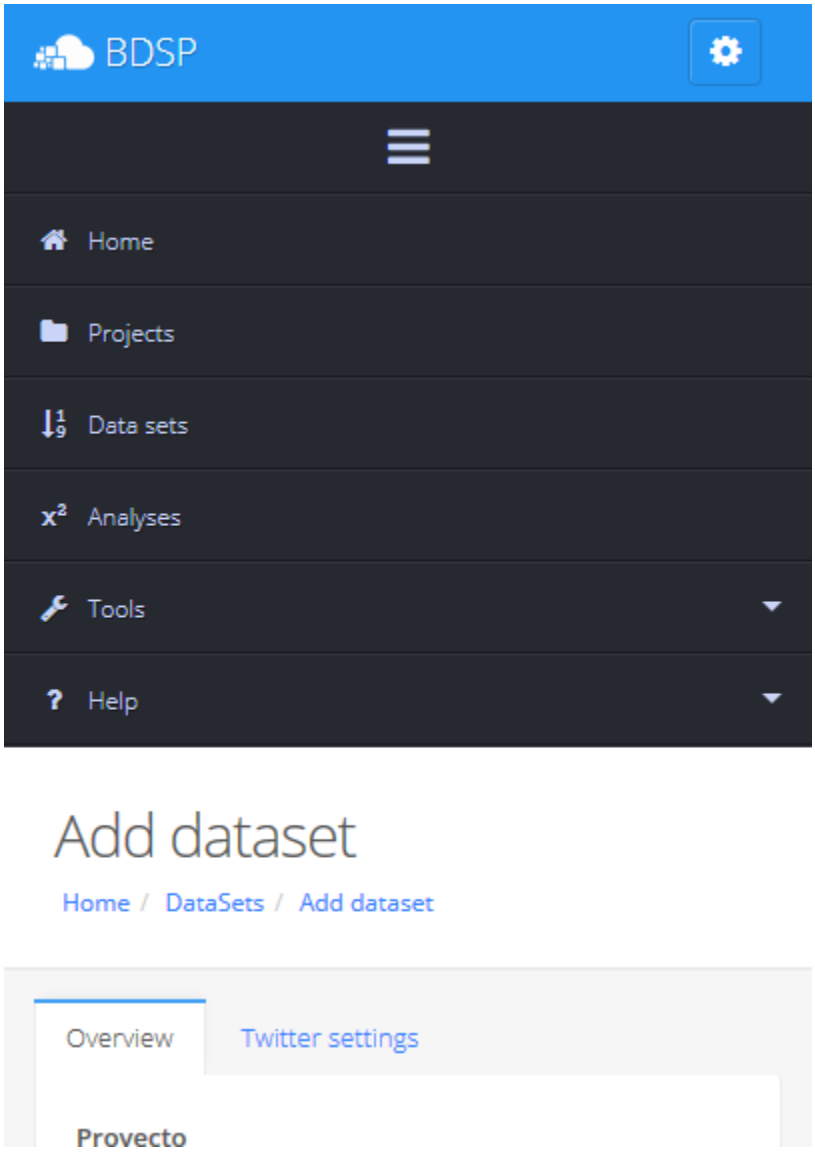

Figura 4.11: Vista del menú de opciones de *BDSP* en un *smartphone*.

tar la autenticación de dos factores (*2FA* del inglés *Two Factor Authentication*), se proporcionará más información sobre *2FA* en la sección 4.6.

La *GUI* cuenta también con un Administrador de Sesiones Activas (ASA) que permite a un usuario ver todas las sesiones que ha iniciado en el sistema, se proporcionará más información sobre el ASA más adelante en este capítulo.

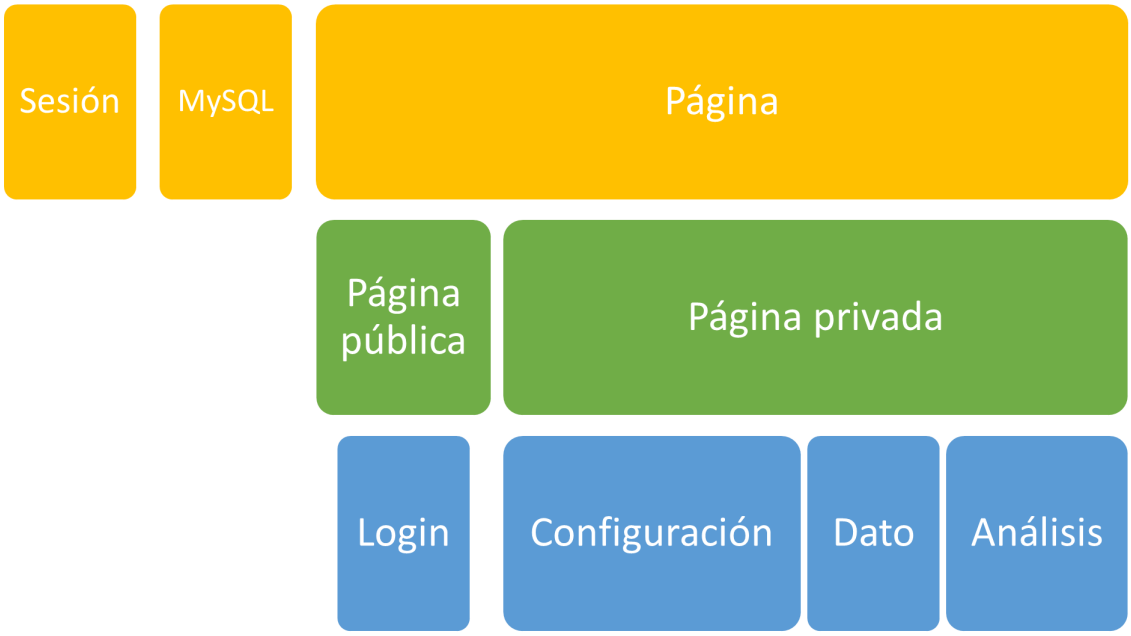

Figura 4.12: Clases de la capa de interfaz gráfica del lado del servidor.

4.5.4. Módulos funcionales y organización de archivos

La capa de interfaz gráfica (*GUI*) se divide en dos secciones de programación: la *GUI* del lado del servidor y la *GUI* del lado del cliente. La *GUI* del lado del servidor consta de elementos programados con *PHP* y *MySQL* cuya función es generar páginas *Web* con código *HTML5* que será utilizado por la *GUI* del lado del cliente. Estas páginas también incorporan scripts programados con *JavaScript*. A continuación se proporcionan los detalles de implementación de estas dos secciones.

GUI del lado del servidor

La función de la *GUI* del lado del servidor es brindar un conjunto de clases y funciones programadas en PHP para facilitar la creación de las páginas *Web* que serán desplegadas en el navegador del usuario. Estas clases se observan en la figura 4.12. Las clases principales son: `Sesión`, `MySQL` y `Página`.

La clase `Sesión` es la encargada de manejar la autenticación del usuario ante

BDSP. Incluye métodos para el inicio y cierre de sesión. Esta clase solo maneja la autenticación mediante llamados a funciones sin generar alguna página *Web*. Esta clase es utilizada por la clase login (como descrito más adelante) para lograr el inicio de sesión en BDSP que si despliega una página Web.

La clase `MySQL` se encarga de facilitar el acceso a la base de datos. Incluye métodos que permiten generar de manera automática el código de consultas `SELECT`, `INSERT` y `UPDATE` de tal manera que las clases que instancían objetos `MySQL` pueden manipular la base de datos sin la necesidad especificar directamente las consultas, lo cual facilita la programación y reduce los errores. Por ejemplo, el siguiente código realiza una consulta a una base de datos previamente abierta:

La clase `Página` se encarga de construir la estructura *HTML* básica de todas las páginas *Web* mostradas por *BDSP*. Esta clase es la encargada de agregar dentro de la estructura *HTML* todos los archivos necesarios para el funcionamiento de la GUI del lado del cliente, como hojas de estilo (con extención .css) y *scripts* en *JavaScript* (con extención .js), por ejemplo, se incluyen los archivos de las bibliotecas jQuery y Bootstrap. La clase `Página` hereda a otras clases tales como `Página pública` y `Página privada` que se describen a continuación.

La clase `Página pública` es la encargada de desplegar el contenido de páginas que no requieren que un usuario haya iniciado sesión en BDSP. Por ejemplo, la página registro de usuario y la página login. Estas dos páginas heredan de la clase `Página pública` para funcionar.

La clase `Página privada` es la encargada de desplegar el contenido de páginas que requieren que un usuario haya iniciado sesión en BDSP, por ejemplo, cada una de las páginas de especificación de una fuente de datos (Twitter, Facebook, etc.) y las páginas de análisis (clasificación, agrupamiento, etc.). Estas páginas heredan de la clase `Página privada` para funcionar. Si se intenta acceder a estas páginas sin que el usuario haya iniciado sesión previamente se redirigirá el navegador a la página de login, impimiendo mostrar el contenido a un usuario no autorizado.

Todas estas clases se encuentran en el directorio:

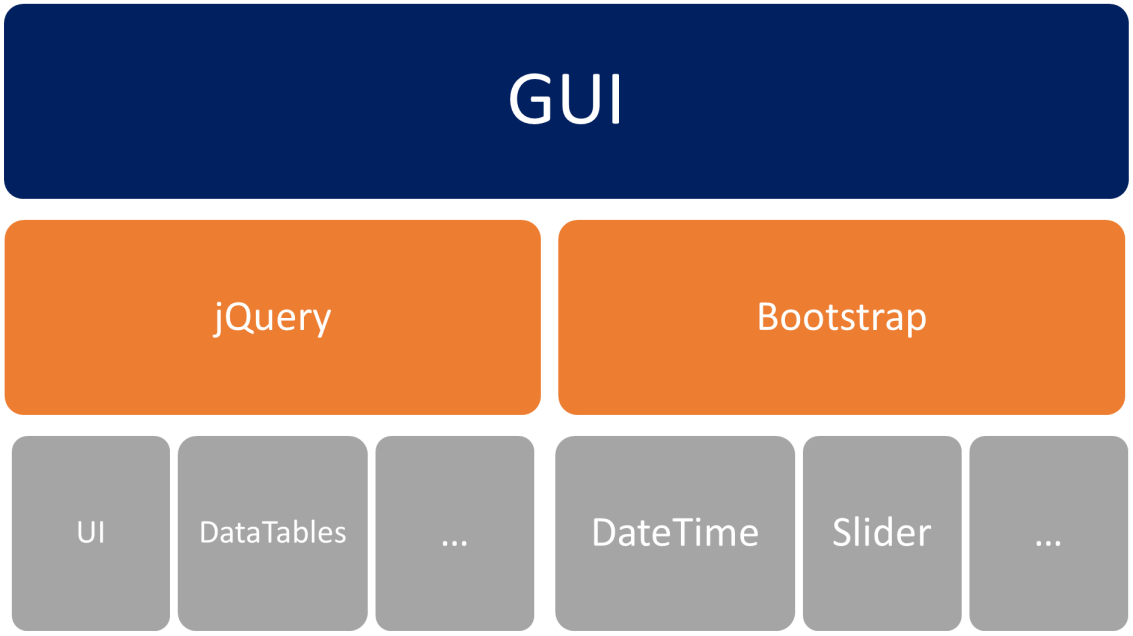

Figura 4.13: Módulos de la capa de interfaz gráfica del lado del cliente.

`__BDSP__/capa-gui/{clase}.php`

donde `__BDSP__` como ya se mencionó, representa el directorio donde se instaló BDSP y `clase` es el nombre de la clase sin espacios y sin acentos.

GUI del lado del cliente

La función de la GUI del lado del cliente es manejar la interacción del usuario con las páginas Web de BDSP. Esta interacción incluye procesos Ajax, validación de formularios y control de la distribución del contenido en la pantalla según el dispositivo que se esté utilizando (Ver sección 4.5.2).

La GUI del lado del cliente se compone de un conjunto de bibliotecas JavaScript que pueden ser basadas en Bootstrap o en jQuery. Estas bibliotecas son incluidas en el archivo HTML por la clase `Página`. Algunas de las bibliotecas JavaScript utilizadas por la GUI se muestran en la figura 4.13.

Las bibliotecas se encuentran en el directorio:

`__BDSP__/capa-gui/cliente/{nombre de biblioteca}`

En este directorio se encuentran los siguientes subdirectorios con los archivos fuentes de todas las bibliotecas JavaScript que utiliza BDSP, e incluyen algunas de las bibliotecas basadas en Bootstrap y jQuery:

autonumeric

bootstrap

bootstrap.datetimepicker

bootstrap.slider

bootstrap.switch

font-awesome

google.prettify

jquery

jquery-ui

jquery.datatables

jquery.easypiechart

jquery.flot

jquery.formvalidation

jquery.gritter

jquery.nanoscroller

jquery.nestable

jquery.select2

jquery.sparkline

Los nombres de estos subdirectorios son los utilizados por la comunidad de desarrollodores y una descripción de las funciones que ofrecen los archivos fuente en cada subdirectorios puede ser consultada en [63, 64].

4.6. Aspectos de seguridad

Para que un usuario pueda hacer uso de *BDSP*, éste debe contar con un nombre de usuario y contraseña los cuales se deben especificar al momento de crear su usuario en el formulario de registro. Los usuarios también deben proporcionar una dirección de correo electrónico que será verificada mediante el envío de un correo electrónico con un enlace de activación. Una cuenta no puede ser utilizada mientras no sea activada.

Cada vez que un usuario inicia sesión en *BDSP*, se crea un registro en la base de datos en donde se almacena la fecha y hora, la dirección *IP* y la ubicación aproximada donde se realizó el inicio de sesión, y la última acción realizada. La ubicación es obtenida mediante el uso de una base de datos de direcciones *IP* llamada *DB-IP*, la cual es actualizada frecuentemente y contiene la ubicación aproximada de más de ocho millones de direcciones *IP* públicas.

BDSP cuenta con un Administrador de Sesiones Activas (ASA) con el cual los usuarios pueden ver todas las sesiones que se han abierto en el sistema utilizando su nombre de usuario. El ASA hace uso de los registros de sesiones creados en la base de datos y muestra toda la información disponible de cada sesión. Un usuario puede en cualquier momento cerrar sesiones remotamente.

Un periodo de inactividad de 15 minutos también ocasionará un cierre de sesión en *BDSP*. Cuando se cierra una sesión, el usuario será direccionado a la página de login en donde tendrá que volver a iniciarla para poder seguir utilizando *BDSP*.

BDSP guarda el historial de acciones realizadas por cada sesión de usuario. Este historial almacena datos como la fecha y hora, acción realizada e identificadores de los objetos afectados, por ejemplo, proyectos, archivos, análisis, *etc.*. En la versión actual este historial está disponible para ser consultado únicamente por administradores de *BDSP*. Los usuarios no tienen acceso a esta información mediante la *GUI*.

BDSP implementa la autenticación de dos factores o *2FA* (del inglés *Two Factor Authentication*), como medida opcional de seguridad. Se ha implementado el algoritmo basado en tiempo de contraseñas de un solo uso o *TOTP* (del inglés *Time-based*

One-time Password Algorithm) el cual permite generar contraseñas que sólo son válidas durante un pequeño periodo de tiempo [65].

Para poder utilizar *2FA*, los usuarios deben contar con una aplicación de generación de contraseñas de un solo uso (GCS) instalada preferentemente en un dispositivo móvil. Las aplicaciones de este tipo son es desarrolladas por terceros y se encuentran en todas las plataformas comerciales existentes: *Windows*, *MacOS*, *Android*, *Windows Phone*, *etc.*.

Para activar *2FA* el usuario debe ingresar el código generado por *BDSP* en su aplicación GCS. Algunas aplicaciones GCS tienen la opción de ingresar este código mediante el escaneo de un código de barras bidimensional. Una vez ingresado este código a la aplicación, ésta generará una contraseña cada 30 segundos.

Si un usuario de *BDSP* habilita *2FA*, una vez iniciado sesión con su nombre de usuario y contraseña, el sistema le pedirá que ingrese la contraseña de un solo uso generada por la aplicación. No podrá ingresar al sistema mientras no complete estos dos pasos para autenticarlo.

Los proyectos, archivos de datos y análisis realizados en *BDSP* sólo pueden ser visualizados por el usuario que los creó. En la *GUI* de la versión actual de *BDSP* no es posible compartir ni transferir proyectos, conjuntos de datos ni análisis con otros usuarios, aunque la *API* sí lo permite por lo que éste puede ser un cambio recomendable para una nueva versión.

4.7. Resumen

En este capítulo se presentó *BDSP*, un sistema Web en el que los usuarios pueden manejar datos y realizar análisis Big Data sobre éstos en cualquier momento y desde cualquier lugar utilizando sólo un dispositivo con conexión a Internet y un navegador.

BDSP está constituido de tres capas: la capa de interfaz gráfica (*GUI*), la capa de análisis de datos (AD) y la capa de manejo de datos (MD).

La *GUI* es la interfaz de un sitio *Web* con un diseño *Web* adaptable que pro-

porciona una experiencia óptima de navegación sin importar el dispositivo que se esté utilizando. Mediante la *GUI*, los usuarios pueden manejar sus datos, así como especificar análisis análisis sobre éstos.

La AD consta de servicios *Web* que encapsulan varios *frameworks* y herramientas existentes para el análisis de datos. Las herramientas que *BDSP* encapsula y ejecuta en paralelo sobre *Hadoop* son *Mahout* y *NLTK*. La *AD* le proporciona algunos parámetros a estos *frameworks* y redirecciona el resultado generado a la *GUI* para que sea desplegado. En la versión actual de *BDSP* la AD puede realizar en total 12 análisis, entre los que se incluyen agrupamiento, clasificación, regresión y análisis de sentimiento.

La MD consta en servicios *Web* que pueden acceder a archivos de datos locales y remotos. Entre los archivos remotos que se pueden manejar están *Dropbox*, *Google Drive*, entre otros. La *MD* almacena los archivos obtenidos en *HDFS*, el sistema de archivos utilizado por *Hadoop*.

BDSP incorpora las siguientes mededidas de seguridad: los proyectos, archivos de datos y análisis realizados sólo pueden ser visualizados por el usuario que los creó. Se ha agregado un Administrador de Sesiones Activas que permite ver todas las sesiones que se han abierto en el sistema por el usuario y es posible cerrar cualquiera de las sesiones de manera remota. Opcionalmente, es posible implementar autenticación de dos factores en la cuenta del usuario.

Capítulo 5

Utilizando *BDSP*

Este capítulo presenta un panorama general de la funcionalidad de *BDSP* desde el punto de vista de los usuarios. Se incluyen dos ejemplos realizados con *BDSP*: un análisis de sentimiento del término *Big Data* analizando *tweets* y un análisis de agrupamiento utilizando un conjunto de datos de flores. Se utilizará el término *análisis* como abreviatura de *análisis de datos* a partir de este momento. El apéndice A presenta otro ejemplo de análisis de datos con *BDSP*.

En general, BDSP es muy fácil de usar, e intuitivo para personas que saben qué tipos de análisis son relevantes al problema que van a resolver. Básicamente sólo se seleccionan opciones con el *mouse* y se especifican nombres de archivos. No hay necesidad de conocer que herramientas de análisis usar y cómo y con qué parámetros.

BDSP se puede mejorar mucho de varias maneras. Particularmente relevante a analistas de datos, sería el añadir funciones de visualización de resultados — ver trabajo futuro en el capítulo 6. Actualmente, BDSP no tiene funciones de visualización: los resultados (la salida) de un análisis son en general mostrados en texto plano. La gráfica mostrada en este capítulo para el segundo ejemplo fue generada por BDSP específicamente para el análisis utilizado; y el usuario no puede configurarlo. Sin embargo, no es difícil diseñar la integración de varias herramientas de visualización.

5.1. Panorama general del uso de *BDSP*

BDSP es un sistema *Web* en el que múltiples usuarios pueden realizar análisis y manejo de archivos de datos de manera concurrente. Todos los usuarios deben registrarse en el sistema antes de poder iniciar una sesión utilizando su nombre de usuario y contraseña. Opcionalmente, los usuarios pueden registrarse utilizando su cuenta de *Facebook* o *Twitter*.

BDSP puede ser utilizado a través de un navegador desde cualquier dispositivo con acceso a Internet. Cuenta con un diseño Web adaptable que le proporciona a los usuarios una experiencia óptima de visualización sin importar el dispositivo que estén utilizando. La *GUI* se adapta al tamaño y capacidades de visualización del dispositivo que se esté utilizando, ya sea un *SmartPhone*, una tableta o una computadora personal.

La figura 5.1 y 5.2 muestran la página principal de *BDSP* que es mostrada una vez que el usuario inicia sesión. Por motivos de legibilidad se ha fragmentado la imagen. En la parte izquierda de la página de inicio se encuentra el menú principal de opciones que incluyen: Inicio, Proyectos, Datos, Análisis, Herramientas y Ayuda. En el sistema, estas opciones se encuentran escritas en inglés y corresponden a *Home*, *Projects*, *Data Sets*, *Analyses*, *Tools* y *Help*.

La página de inicio muestra un resumen del estado de los proyectos creados por el usuario, así como de tareas de análisis y manejo de archivos de datos. Esta página también muestra, en la parte inferior, notificaciones sobre el estado de tareas que anteriormente han sido realizadas, o deben ser realizadas, por el usuario.

Proyectos (*Projects* en la figura 5.1). Los archivos de datos y análisis sobre éstos se organizan en proyectos. Cada proyecto puede tener múltiples archivos de datos y múltiples análisis. Las opciones que se puden realizar sobre un proyecto son: crear un archivo de datos, especificar un análisis sobre un archivo, eliminar un archivo de datos o análisis, desplegar el resultado de un análisis, nombrar o renombrar un análisis con un nombre fácil de recordar. En la página de inicio, *Projects in progress* (en español,

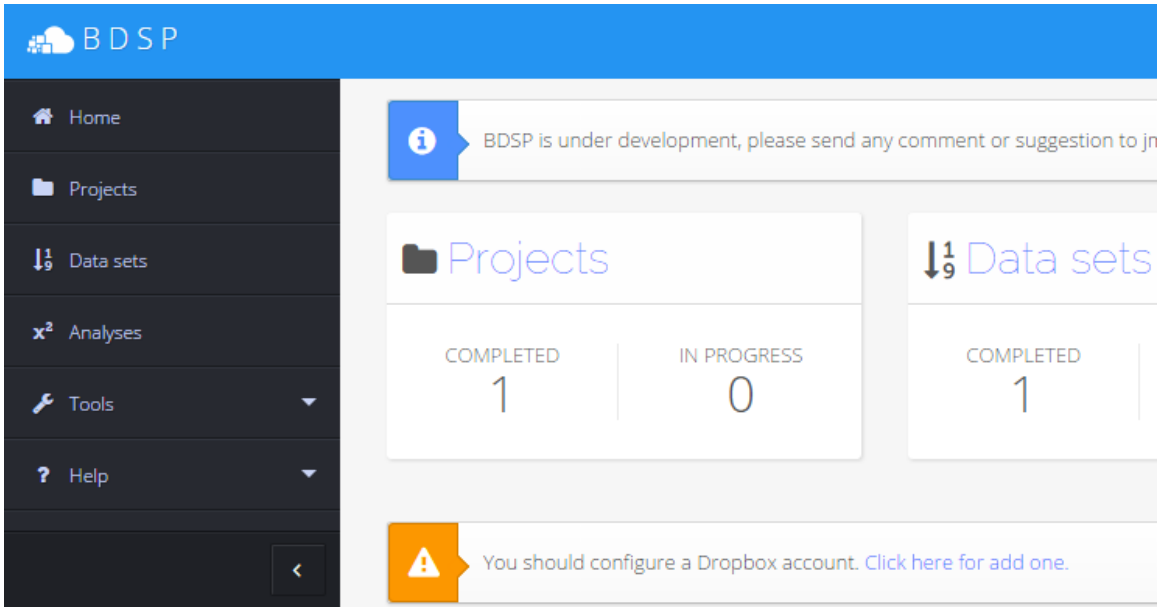

Figura 5.1: Parte izquierda de la página principal de *BDSP*.

proyectos en progreso) indica el número de proyectos que tienen adquisiciones de datos o análisis que aún no se han terminado de procesar.

Datos (o archivos de datos, *Data sets* en la figura 5.1). Son los archivos que el usuario subió a BDSP a partir de una copia de un archivo local o remoto en sistemas *Web* tales como *Dropbox* y *Google Drive*, *tweets* en *Twitter* o comentarios en *Facebook*, entre otros. Las funciones que se pueden realizar sobre los datos son: crear, eliminar, renombrar archivos de datos, tal y como ocurre con los proyectos. Cuando una función de datos es invocada sin haber creado o seleccionado un proyecto, este nuevo dato es asignado al *Default Project* (en español, proyecto por defecto), el cual puede ser eventualmente renombrado. Crear un archivo de datos involucra crear una copia de un archivo fuente (local o remoto) en el sistema de archivos *HDFS* de *BDSP*.

Analisis (*Analyses* en la figura 5.1). Las funciones relacionadas con un análisis son: crear, eliminar, nombrar y renombrar un análisis. Un análisis puede ser realizado sólo sobre un archivo de datos existente. No es posible especificar análisis en el vacío. El resultado de un análisis es almacenado en *HDFS* y puede ser visualizado una vez

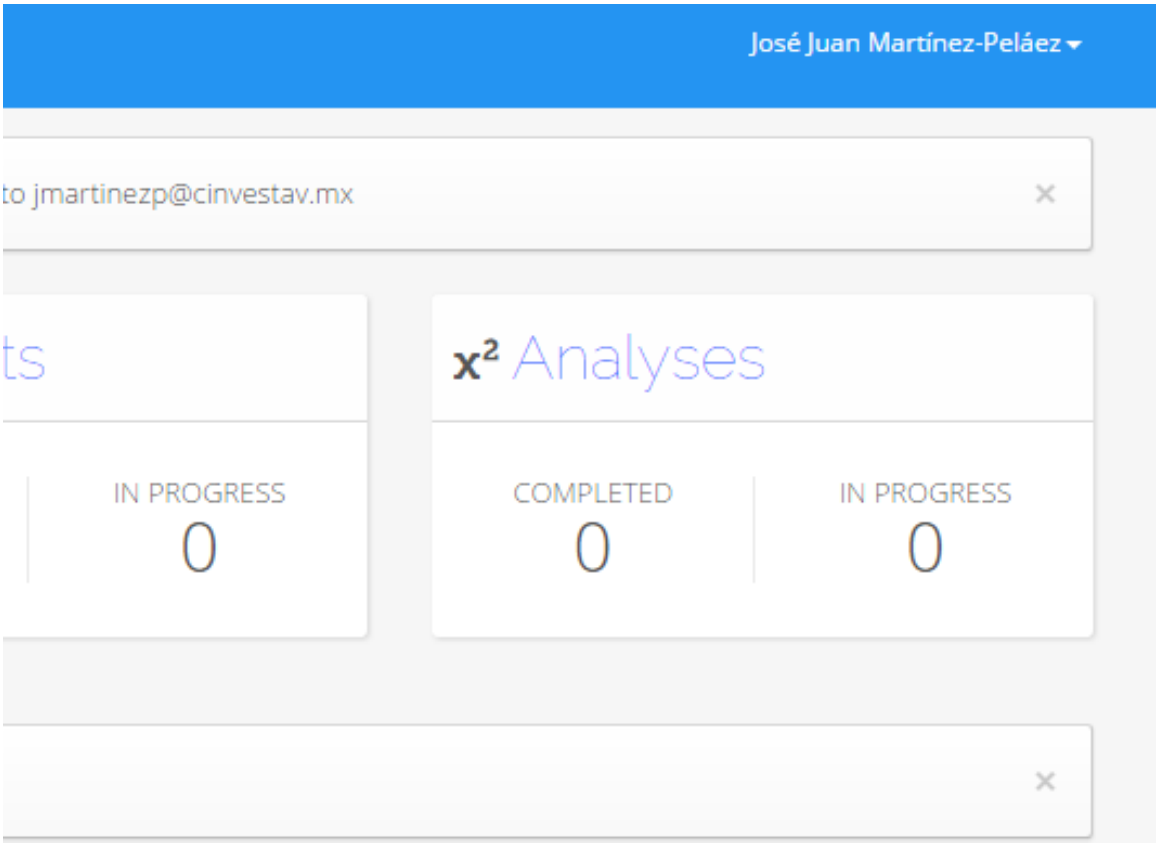

Figura 5.2: Parte derecha de la página principal de *BDSP*.

que éste ha finalizado. El archivo de resultado debe ser eliminado explícitamente, es decir, eliminar un análisis de *BDSP* no elimina el archivo de resultados. La versión actual de *BDSP* ofrece los siguientes análisis: regresión (simple, múltiple, lineal y no lineal), clasificación (*C4.5*, *ID3*), Agrupamiento (*K-Means*, *Fuzzy C-Means*), muestreo (Gibbs, Metropolis-Hastings, Monte Carlo) y análisis de sentimiento (mediante el uso de bolsas de palabras).

BDSP cuenta con dos herramientas que permiten configurar y controlar la cuenta del usuario. Una de estas herramientas permite agregar o eliminar cuentas de sistemas externos tal como *Dropbox* o *Google Drive*, de tal manera que *BDSP* pueda recuperar archivos almacenados en estos sistemas en nombre del usuario al momento de crear un archivo de datos en BDSP. La recuperación de archivos se basa en el uso de *tokens*

de autorización que son proporcionados por los sistemas externos al momento que el usuario los agrega. En ningún momento *BDSP* almacenará el nombre de usuario o contraseña del usuario en estos servicios externos.

Otra herramienta es el Administrador de Sesiones, el cual muestra todas las sesiones activas realizados por el usuario en todos los dispositivos, es decir, muestra todos los inicios de sesión realizados por el usuario. Desde esta herramienta, el usuario puede cerrar cualquier sesión de manera remota.

La opción de ayuda en esta versión de *BDSP* aún no contiene nada pero en una versión futura puede ser utilizada para proporcionar ayuda en temas relacionados tanto en *Big Data* como en el funcionamiento de *BDSP*. Puede contener tutoriales en línea, una sección con las preguntas frecuentes y servir como un canal de comunicación para enviar preguntas o comentarios al administrador de *BDSP*.

5.2. Análisis de sentimiento

Se recordará que el análisis de sentimiento es el estudio computacional de la opinión de las personas con el fin de determinar sus actitudes y emociones ante ciertos temas o eventos. El objetivo es identificar el sentir de las personas a través de sus opiniones, y clasificarlas de acuerdo a su polaridad [33]. Una opinión puede ser clasificada como positiva o negativa. Los tipos más importantes de análisis de sentimientos son (los demás son descritos en [33]):

- Clasificación de sentimiento. Realiza una clasificación de un conjunto de opiniones en tres categorías: positivas, negativas o neutrales. Puede ser una tarea compleja cuando las opiniones se encuentran en múltiples idiomas o provienen de varios dominios, como biología, sociología, *etc.*

- Clasificación de subjetividad. Determina si una oración es subjetiva u objetiva. Una oración objetiva contiene información imparcial, mientras que una oración subjetiva contiene información de carácter personal como opiniones.

- Resumen de opinión. Permite extraer las características principales que son compartidas por uno o más documentos y el sentimiento acerca de estas características.

- Recuperación de opinión. Permite extraer documentos que expresan cierta opinión sobre la consulta realizada.

El análisis de sentimiento tiene varios enfoques, el basado en lexicón y el basado en *corpus*.

El enfoque basado en lexicón utiliza una colección de términos conocidos, frases y hasta regionalismos. Este enfoque también es conocido como enfoque basado en diccionarios y utiliza un conjunto inicial de términos que generalmente son recolectados y anotados de manera manual con una categoría (sentimiento) positiva, negativa o neutral. Este conjunto inicial crece al incluir manualmente sinónimos y antónimos de las palabras contenidas. Este diccionario es conocido como bolsa de palabras en la literatura [33]. La principal desventaja del uso de bolsas de palabras es la dificultad de procesar textos con información específica de un determinado contexto o dominio de información, ya que las bolsas de palabras tienden a ser muy generales o muy específicas.

El enfoque basado en *corpus* está basado en el uso de diccionarios para un dominio o contexto en particular. Estos diccionarios pueden ser generados por un conjunto de términos semillas, proporcionados por los usuarios, y posteriormente son extendidos con la búsqueda de palabras relacionadas a estos términos. Se utilizan métodos estadísticos para determinar qué términos adicionales se agregarán. Cada término tiene también un valor de sentimiento positivo, negativo o neutral.

Se han creado algunos diccionarios o bolsas de palabras que están disponibles en Internet para ser utilizadas sin costo. Por ejemplo, *SentiWordNet*, una extención del diccionario *WordNet*, el cual es una base léxica muy grande de términos en inglés. Este diccionario contiene sustantivos, verbos, adjetivos y adverbios que están agrupados en categorías llamandas *synsets*. Cada una de las palabras agrupadas en el *synsets*

tienen un significado similar, diferente al significado de las palabras agrupadas en *synsets* diferentes.

Descripción del problema

Supóngase que un usuario necesita conocer qué piensan las personas sobre *Big Data*. Para este propósito, el usuario puede analizar *tweets* que contengan el término *Big Data* para después poder establecer la polaridad de estos *tweets*.

Obtención de datos

En *BDSP*, el usuario debe añadir un nuevo archivo de datos con los tweets que contengan el término *Big Data* obtenidos desde Twitter. La figura 5.3 muestra el formulario mostrado por *BDSP* que el usuario debe llenar con el fin de programar la adquisición de los datos. Estos datos serán almacenados en *HDFS* de *BDSP*. Se puede acceder a estos formularios seleccionando la opción *Data Sets/Add Data Set*.

El formulario solicita al usuario seleccionar una fuente de datos (*Twitter*), el término de búsqueda (*Big Data*) y la cantidad de tweets que desea recuperar. De manera opcional el usuario también puede especificar que sólo desea recuperar *tweets* que han sido publicados en un periodo específico de tiempo o indicar un lenguaje en particular.

BDSP también soporta extraer archivos de datos desde *Facebook*, *Dropbox*, *Google Drive*, archivos locales o archivos que pueden ser descargados desde una *URL*. Si el usuario ha recolectado los tweets utilizando alguna otra herramienta, puede subirlos a *BDSP* utilizando el formato *.CSV* o *.TXT* para analizarlos.

Análisis de datos

Una vez que el archivo de datos se encuentra disponible, puede serle asignado/programado un análisis de sentimiento. Para esto, el usuario debe seleccionar la opción *Analyses/Add Analysis*. La figura 5.4 muestra el formulario que debe ser llenado por el usuario para programar el análisis. Sin importar el tipo de análisis, el formulario

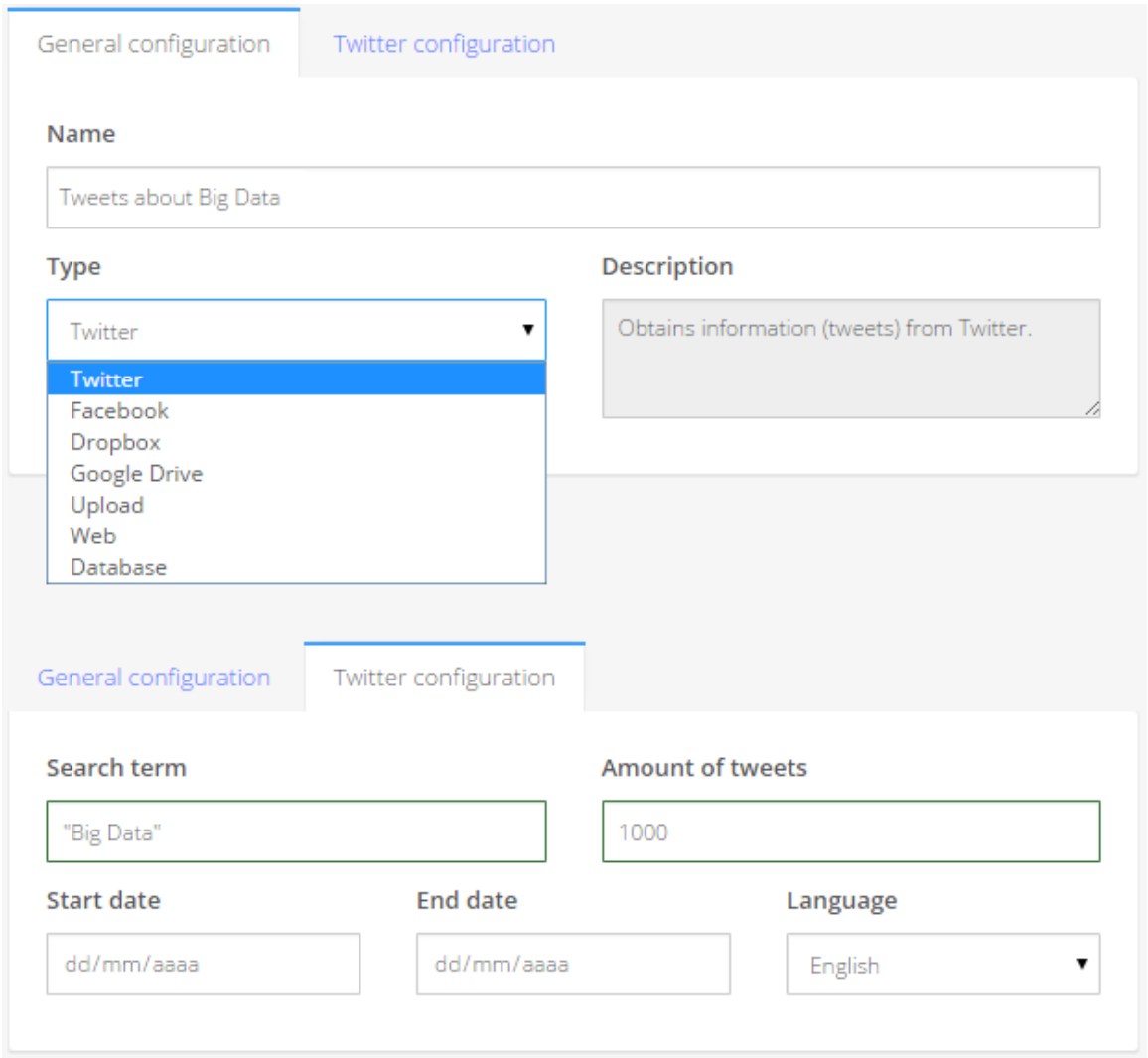

Figura 5.3: Formulario de configuración del tipo de dato *Twitter*.

solicita el nombre con el que se identificará al análisis, seleccionar el archivo de datos que se analizará y especificar el tipo de análisis a realizar. También es necesario configurar los parámetros específicos del análisis. Para este ejemplo, se debe seleccionar la bolsa de palabras *SentiWordNet* e indicar que el campo *Tweet* será el que se analizará.

Resultados

Para este ejemplo, el análisis de sentimiento produjo los siguientes resultados de polaridad para el término *Big Data*: 70.3 % positive 22.4 % neutral 7.3 % negativo.

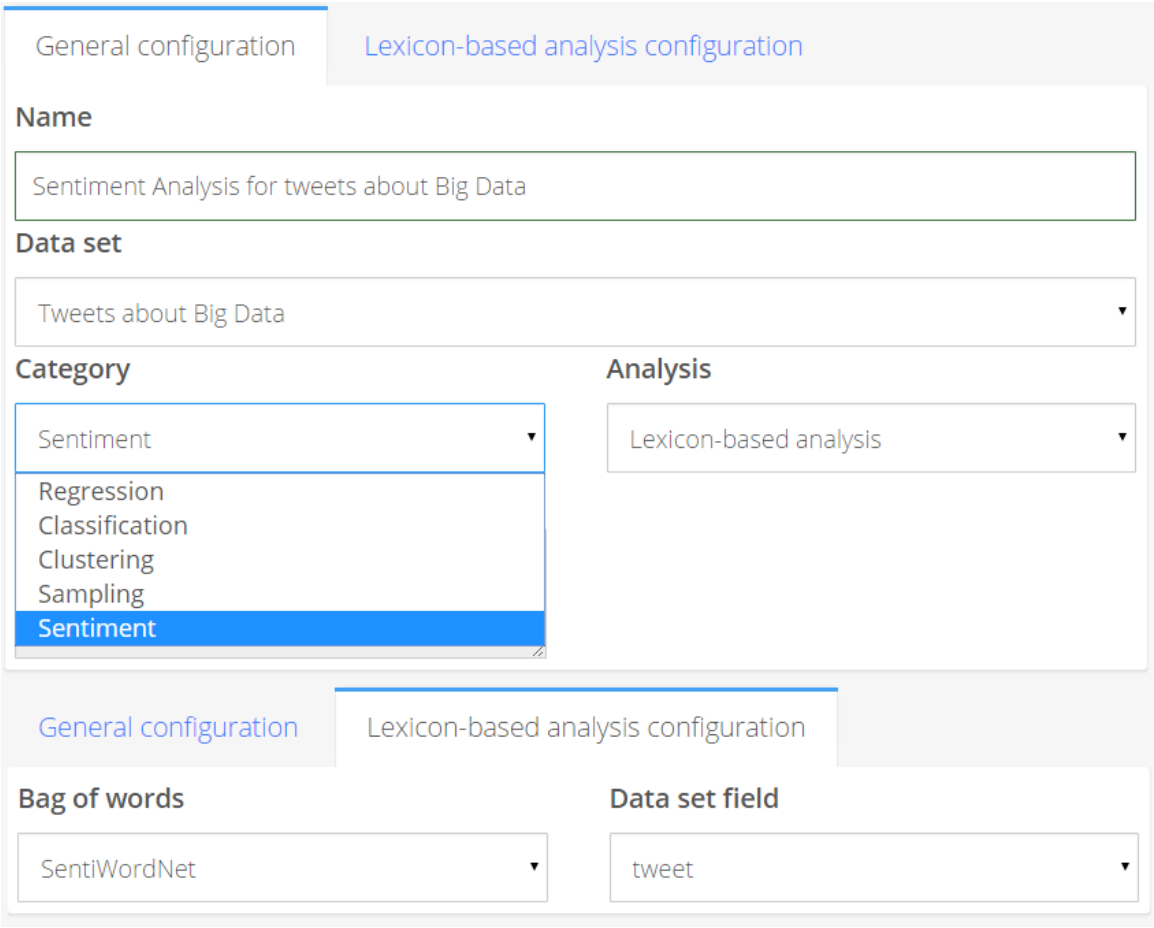

Figura 5.4: Formulario de configuración del análisis de sentimiento.

5.3. Análisis de agrupamiento

Se recordará que el análisis de agrupamiento o *clustering* divide un conjunto de datos en grupos que son significativos de alguna manera, utilizando sólo la información en los datos de entrada.

El objetivo del análisis de agrupamiento es que todos los objetos de un grupo sean, de alguna manera, similares o relacionados entre sí y diferentes de los objetos de otros grupos [30].

El *clustering* algunas veces sólo es un punto inicial que es de utilidad para otros fines tales como el resumen de datos. Este tipo de análisis es usado en múltiples áreas tales como en mercadotécnica, donde es utilizado para el agrupamiento de clientes con comportamiento o características similares; en biología para clasificar plantas y animales según sus características; en ciencias de la tierra para el estudio de los terremotos, en donde a partir de los epicentros se puede identificar zonas de riesgo [66].

K-means es uno de los algoritmos de *clustering* más simples. Este algoritmo sigue una manera sencilla de agrupar un conjunto de datos (de valores numéricos) en un número específico de grupos. Este número de grupos se conoce como K. La idea principal es definir un centroide para cada grupo y asociar cada dato al centroide más cercano. El siguiente paso es recalcular los K centroides a partir de los datos que se le han asignado. Con estos nuevos K centroides se deben reasignar todos los datos a su centroide más cercano. El algoritmo se repite hasta que los centroides dejan de moverse.

Descripción del problema

Suponga que un usuario requiere identificar los grupos de flores contenidos en el conjunto de datos Iris [67]. Este conjunto de datos es multivariable (multiatributo) y fue propuesto por Ronald Fisher. El conjunto de datos consta de 50 muestras de tres tipos de flores iris (Iris setosa, Iris virginica e Iris versicolor). Se incluyen cuatro medidas (atributos) para cada muestra: el largo y ancho en centrímetos del sépalo y del pétalo.

Obtención de datos

Para este ejemplo se asumirá que el usuario tiene el conjunto de datos Iris almacenado en su cuenta de *Dropbox*. Con el fin de analizarlo, el usuario debe crear un

nuevo conjunto de datos en *BDSP*, copiándolo al *HDFS* de *BDSP*. Para esto, debe seleccionar la opción *Data Set/Add Data Set/Dropbox* (esta última en lugar de seleccionar *Twitter* como se hizo en el ejemplo anterior). Asumiendo que *BDSP* ha sido configurado para acceder a los archivos de la cuenta de *Dropbox* del usuario, *BDSP* mostrará los archivos contenidos en *Dropbox* como se muestra en la figura 5.5.

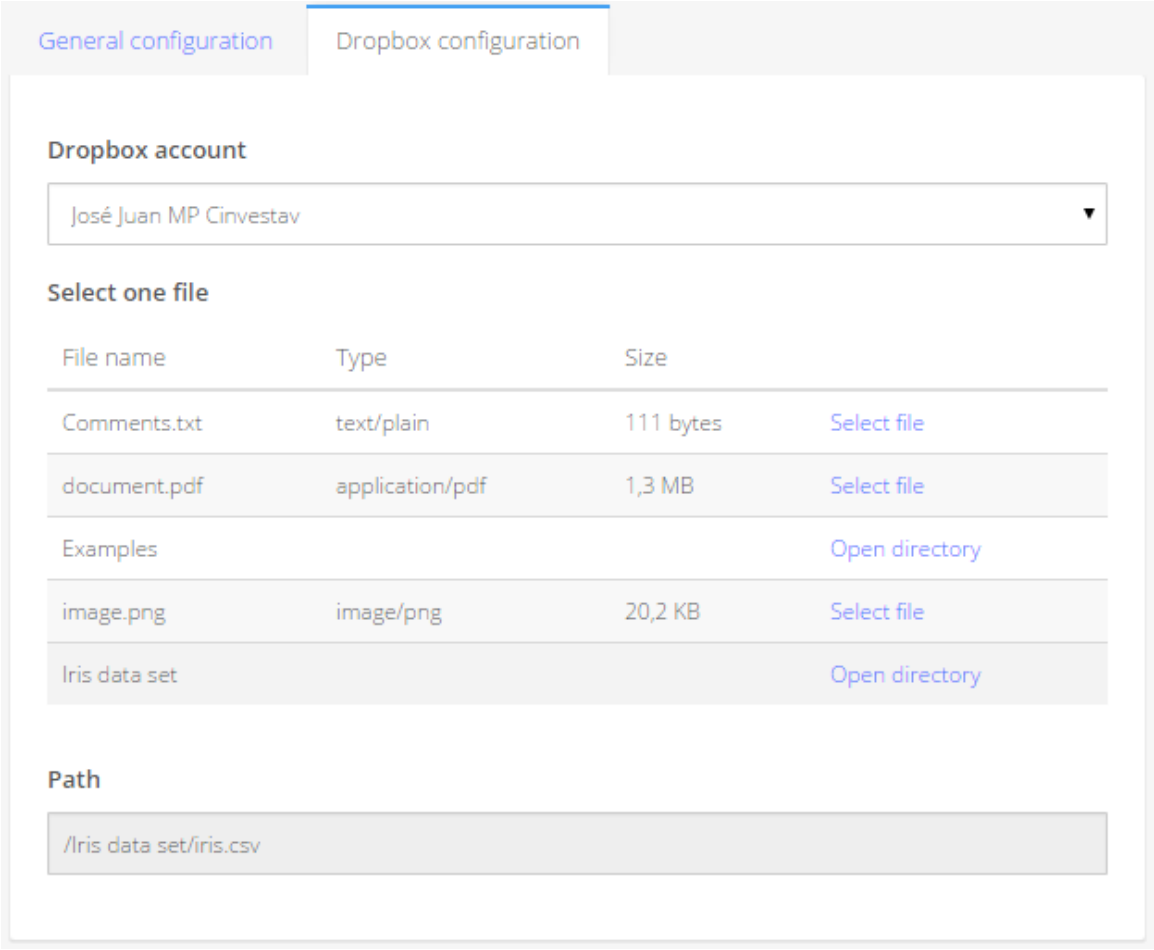

Figura 5.5: Selección del archivo de datos en *Dropbox* mediante *BDSP*.

El usuario puede navegar por el álbol de directorios de *Dropbox* con el fin de seleccionar el archivo correspondiente (iris.csv en este caso). El usuario puede configurar múltiples cuentas de *Dropbox* en caso de ser neceario.

Análisis de datos

De igual manera en que se realizó en el ejemplo anterior, un análisis de agrupamiento utilizando *K-means* puede ser asignado en *BDSP* seleccionando la opción *Analyses/Add Analysis* y después seleccionar *K-means*. Una vez que el usuario realizó esto, debe proporcionar la información solicitada que se muestra en el formulario de la figura 5.6. En este formulario se debe especificar el número K de grupos y los valores que serán utilizados por *K-mean*.

En el archivo de datos Iris, al igual que en la mayoría de archivos de datos en formato .CSV, el primer renglón es el encabezado que contiene el nombre de las columnas o atributos. A continuación se muestran los primeros cinco registros de este repositorio.

sepal length,sepal width,petal length,petal width,class
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa

La figura 5.6, muestra que el usuario ha configurado el análisis utilizando los valores del largo y ancho del cépalo. Estos valores son usados para generar el eje X y Y en el resultado que se muestra en la figura 5.7.

Resultados

Tenga en cuenta que el usuario puede configurar el análisis de agrupamiento con K-means utilizando cualquier combinación de dos atributos disponibles en el archivo de datos. Sin embargo, los valores del sépalo (ancho y largo) producen el mejor agrupamiento en los 3 grupos distintos de flores, de acuerdo a la clase conocida de cada flor dada como el quinto campo en cada línea en el conjunto de datos Iris.

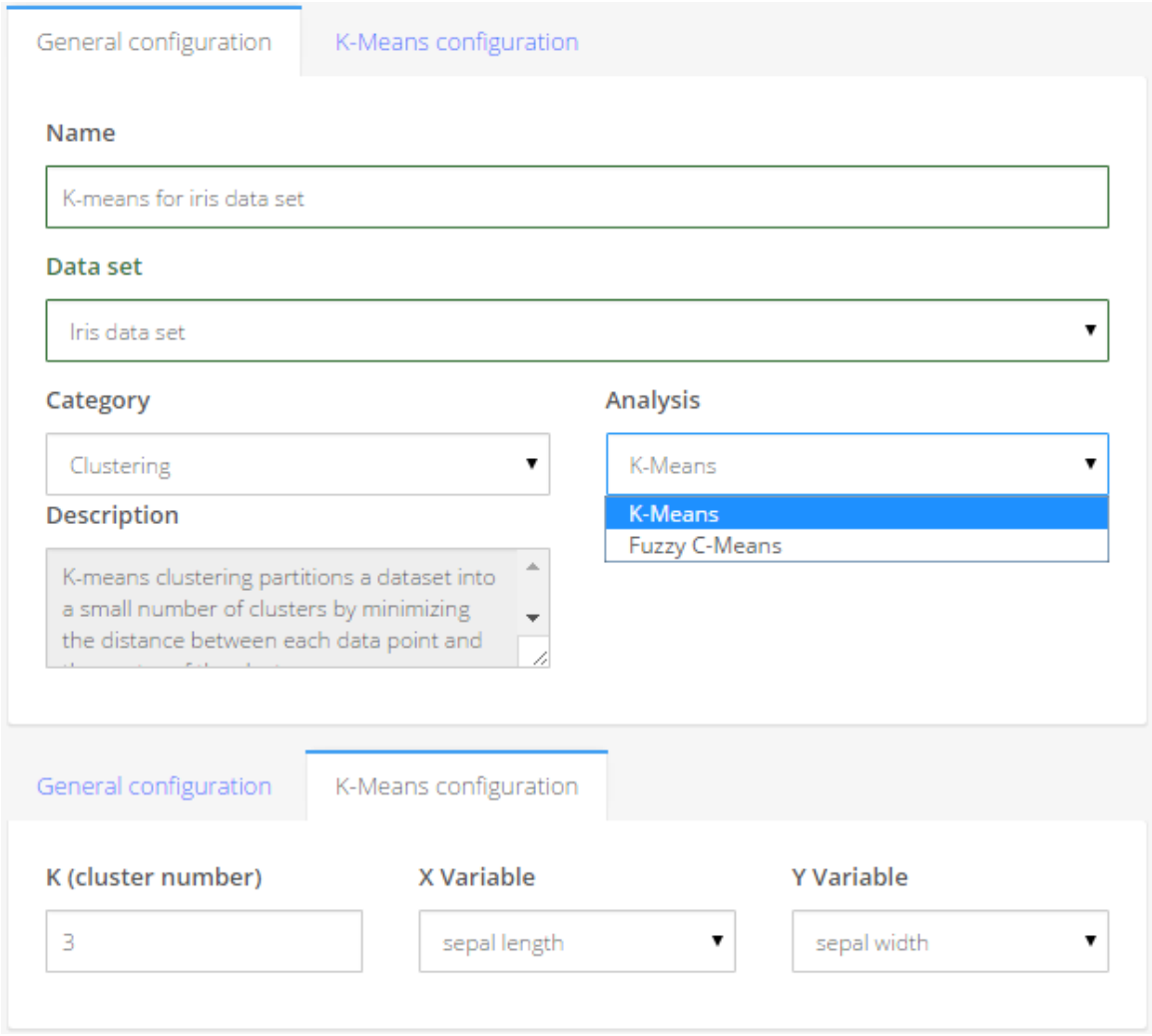

Figura 5.6: Formulario de configuración del análisis *K-means*.

5.4. Configuración de la cuenta de usuario

Antes de utilizar BDSP, todos los usuarios deben crear una cuenta en el sistema. Para ello se debe rellenar el formulario de registro donde se proporciona un nombre de usuario, un correo electrónico y la contraseña. Un correo electrónico será enviado para verificar que la cuenta de correo electrónico sea válida. La figura 5.8 muestra el formulario de registro en *BDSP*. Una vez que el usuario se ha registrado y ha confirmado su cuenta mediante la apertura del enlace enviado a su correo electrónico

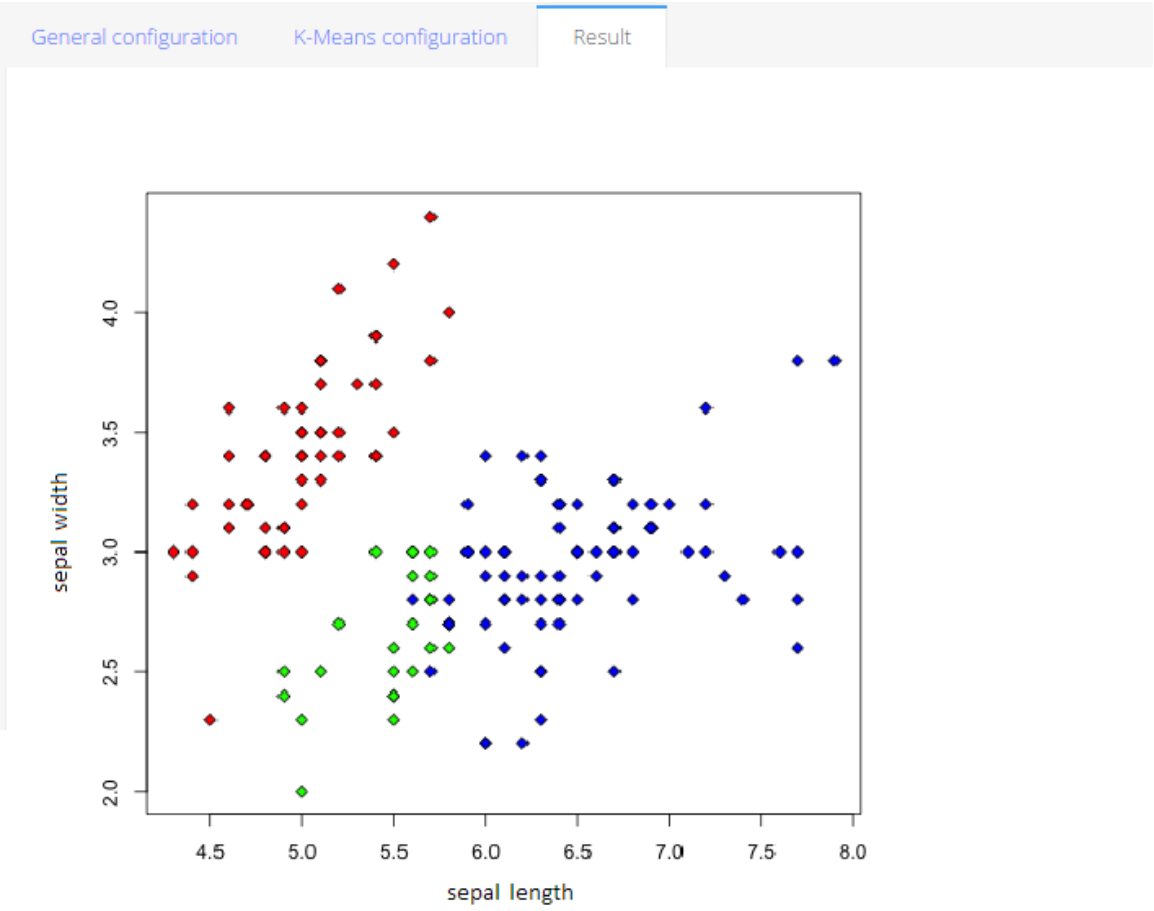

Figura 5.7: Resultado del análisis *K-means*.

podrá iniciar sesión en *BDSP*.

BDSP permite realizar configuraciones básicas sobre al perfil de usuario tales como modificar el nombre, cambiar el correo electrónico o la contraseña y habilitar o deshabilitar la autenticación de dos factores. Para configurar el perfil de usuario se debe dar click en el nombre del usuario que se encuentra ubicado en la esquina superior derecha de la pantalla donde se desplegará un menú en el cual se debe seleccionar la opción *My account*. Este menú se muestra en la figura 5.9.

Para cambiar la contraseña se debe proporcionar la contraseña actual del usuario y escribir por duplicado la nueva contraseña. En la versión actual de *BDSP* no se ha implementado ninguna política de contraseñas por lo que cualquier contraseña de

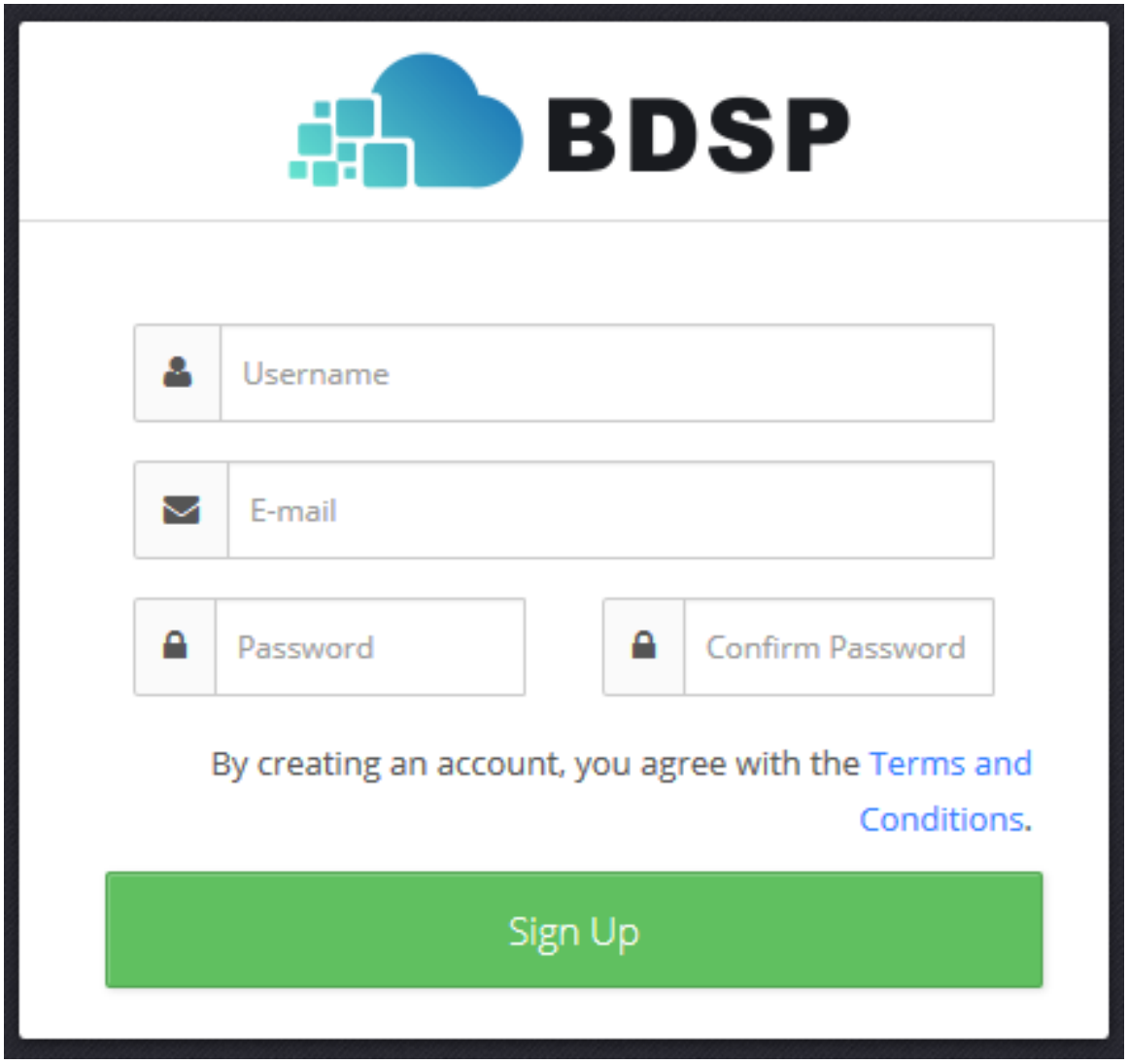

Figura 5.8: Formulario de registro de usuarios en *BDSP*.

un carácter o más será aceptada. El formulario para cambiar la contraseña puede observarse en la figura 5.10.

Como se mencionó en el capítulo 4, la autenticación de dos factores es un mecanismo de seguridad que es recomendable habilitar en la cuenta del usuario. Para esto es necesario que instale en su dispositivo móvil un programa para la generación de contraseñas de un solo uso. Existen muchos programas desarrollados por terceros, por ejemplo, para la plataforma Android y para la plataforma *IOs* puede instalar Au-

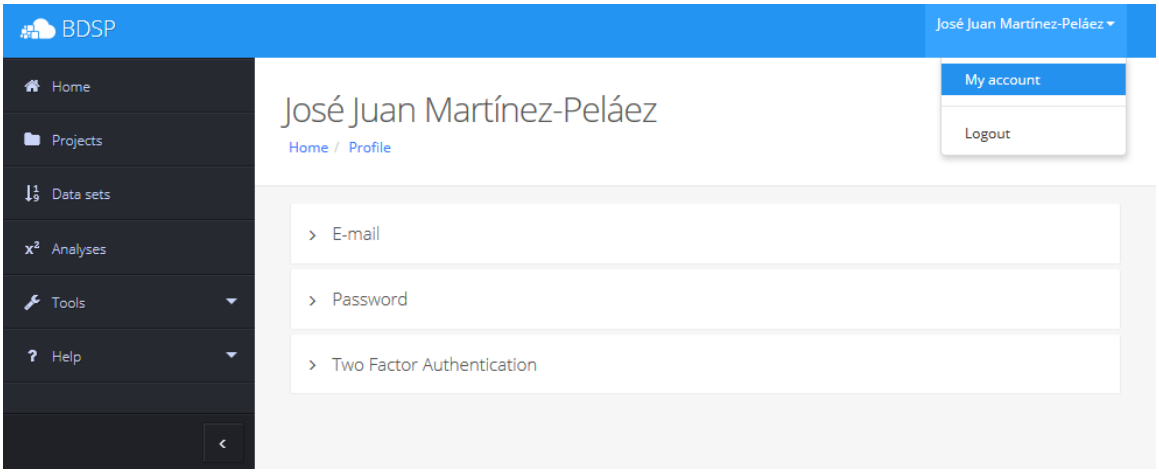

Figura 5.9: Configuración del perfil de usuario en *BDSP*.

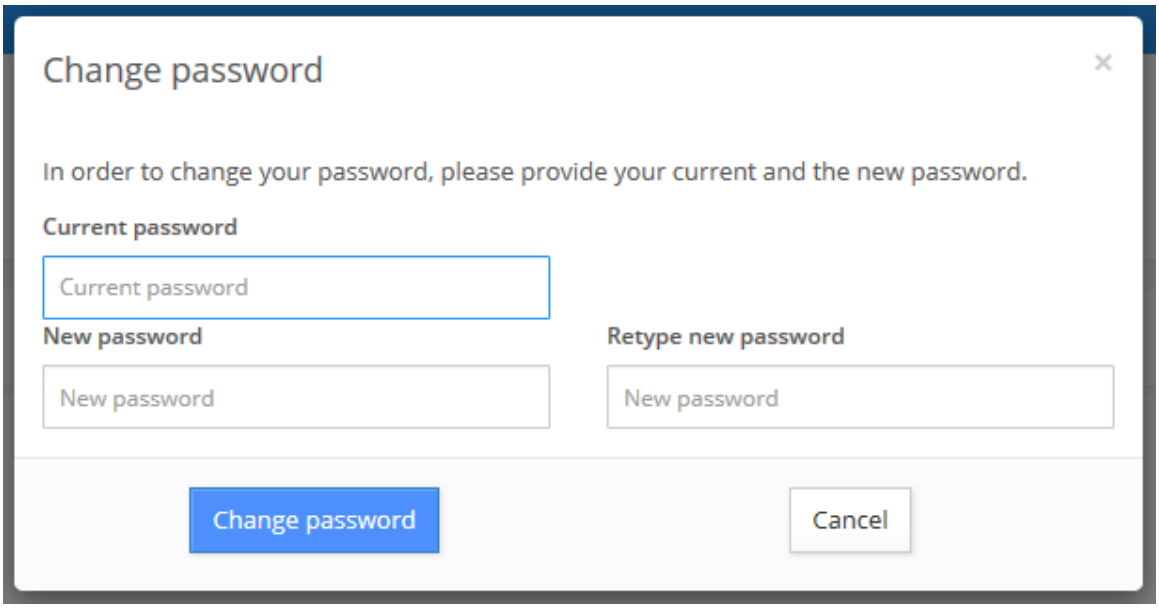

Figura 5.10: Formulario para cambiar la contraseña del usuario.

tenticador desarrollado por Google, para *Windows Phone*, *Microsoft* ha desarrollado una aplicación que también tiene el nombre de Autenticador. En plataformas como *Windows*, *Linux* y *MacOS* también se han desarrollado programas para la generación de este tipo de contraseñas.

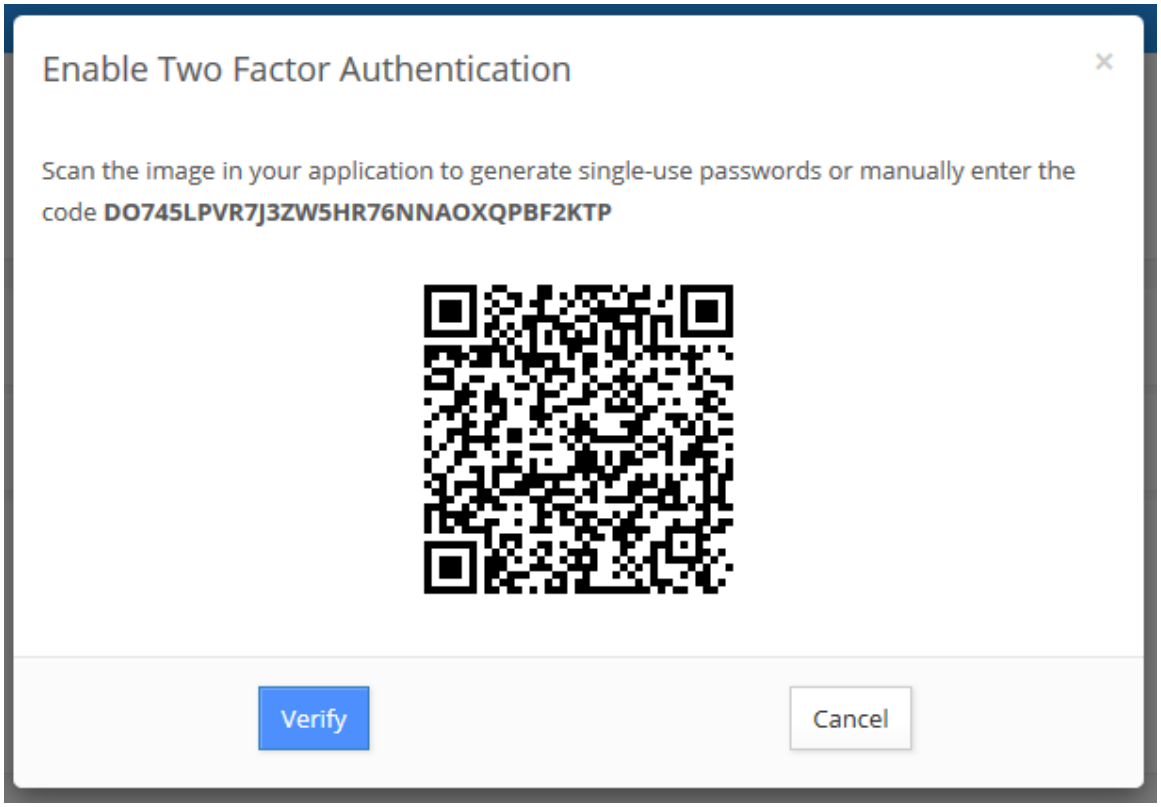

Figura 5.11: Formulario para habilitar la autenticación de dos factores en la cuenta de un usuarios en BDSP.

Una vez que ha instalado la aplicación de generación de contraseñas, debe dar click en el botón *Enable TFA* y se mostrará el formulario que se muestra en la figura 5.11. En este formulario se muestra el código que debe de ingresar en el campo clave secreta de su aplicación. Si la aplicación lo permite puede escanear el código *QR* que se muestra en el formulario. Deberá verificar el código generado por la aplicación antes de habilitarlo en *BDSP*.

Una vez que se ha habilitado *TFA*, cada vez que el usuario inicie sesión en *BDSP* además del nombre de usuario y contraseña se solicitará el código de un solo uso generado por la aplicación.

5.5. Resumen

Este capítulo presentó un panorama general de la funcionalidad de *BDSP* desde el punto de vista de los usuarios. *BDSP* es un sistema *Web* en el que múltiples usuarios pueden realizar análisis y administración de archivos de datos de manera concurrente.

Los archivos de datos son los archivos que el usuario subió a *BDSP* a partir de una copia de un archivo local o remoto en sistemas *Web* tales como *Dropbox* y *Google Drive*, *tweets* en *Twitter* o comentarios en *Facebook*, entre otros. Las funciones que se pueden realizar sobre los datos son: crear, eliminar, renombrar archivos de datos.

Los análisis pueden ser especificados sólo sobre un archivo de datos existente. El resultado de un análisis es almacenado en *HDFS* y puede ser visualizado una vez que éste ha finalizado. Las funciones relacionadas con un análisis son: crear, eliminar, nombrar y renombrar un análisis.

Los archivos de datos y análisis sobre éstos se organizan en proyectos. Cada proyecto puede tener múltiples archivos de datos y múltiples análisis. Las opciones que se pueden realizar sobre un proyecto son: crear un archivo de datos, especificar un análisis sobre un archivo, eliminar un archivo de datos o análisis, desplegar el resultado de un análisis, nombrar o renombrar un análisis con un nombre fácil de recordar.

Para obtener cualquier archivo de datos se deben llenar los formularios de la opción *Data Sets/Add Data Set*. En esta opción se muestra un formulario general en donde se especifica el nombre del archivo de datos a crear, el proyecto al que se asignará y origen de los datos. Adicionalmente hay que proporcionar los parámetros específicos de cada origen de dato.

Para especificar cualquier análisis se deben llenar los formularios de la opción *Analyses/Add Analysis*. En esta opción se muestra un formulario general en donde se especifica el nombre del análisis a realizar, el archivo de datos que se analizará y el proyecto al que se asignará el archivo de resultados. Adicionalmente hay que proporcionar los parámetros específicos de cada análisis.

Capítulo 6

Conclusiones

Esta tesis presentó *BDSP* (del inglés *Big Data Start Platform*), un sistema *Web* en el que los usuarios pueden administrar sus datos y especificar análisis de *Big Data* sobre éstos. *BDSP* fue creado a partir de la integración de herramientas y *frameworks* existentes para el análisis de datos. El sistema cuenta con los siguientes componentes:

- un servidor *Web* que ejecuta servicios en las capas que constituyen *BDSP*
- un servidor de base de datos que almacena la información de los usuarios como contraseñas y *tokens* de acceso a los servicios de terceros
- un *cluster* de procesamiento basado en *Hadoop* donde los datos son procesados y almacenados

Las capas de *BDSP* son: la *GUI*, la capa de manejo de datos y la capa de análisis de datos. Cada una de estas capas puede ser extendida o remplazada. *BDSP* está motivado por la necesidad de crear prototipos rápidos de proyectos de *Big Data*. Puede ser modificado según se necesite, y usado como un vehículo de entrenamiento para el análisis de datos y para el desarrollo de *software* para *Big Data*.

La *GUI* es la capa de *software* que permite que *BDSP* pueda ser accedido desde cualquier dispositivo con conexión a Internet mediante el uso de un navegador compatible con *HTML5*. Esta capa permite configurar y administrar proyectos, conjutos

de datos y especificar análisis de manera sencilla. Cuenta con un Diseño Adaptable que facilita el uso de *BDSP* sin importar las características del dispositivo que se esté utilizando.

La capa de manejo de datos es la encargada de adquirir datos locales (localizados en el dispositivo del usuario) o remotos desde distintas fuentes *Web* tales como *Dropbox*, *Google Drive*, *Twitter* y *Facebook*. Los datos obtenidos son almacenados como un archivo en *HDFS*, el sistema de archivos distribuidos con tolerancia a fallas utilizado por *Hadoop*.

La capa de análisis es la responsable de ejecutar los algoritmos de análisis disponibles sobre los archivos de datos. La función de esta capa es abstraer los detalles de implementación de cada *framework*, o biblioteca, de análisis y proporcionar mecanismos de ejecución similares. La capa de análisis utiliza algunos de los análisis existentes en *Mahout* y *NLTK* y los ejecuta en paralelo utilizando *Hadoop*. *BDSP* actualmente es capaz de realizar 12 análisis de datos distintos entre los que se incluyen análisis de: regresión (simple, múltiple, lineal y no lineal), clasificación (*C4.5*, *ID3*), agrupamiento (*K-Means*, *Fuzzy C-Means*), muestreo (Gibbs, Metropolis-Hastings, Monte Carlo) y análisis de sentimiento (bolsa de palabras).

Todas estas capas pueden ejecutarse en un solo servidor o pueden ser ejecutadas en servidores distintos, según los requerimientos de cada proyecto.

6.1. Limitaciones y trabajo futuro

BDSP es una plataforma base, la versión que se presenta en esta tesis puede ser mejorada de muchas maneras. A continuación se presentan algunas ideas que mejorarían su funcionalidad.

6.1.1. Interfaz

BDSP soporta un número limitado de fuentes remotas de datos. Sería conveniente añadir otras fuentes y otros tipos de datos, como fotografías o video. Por ejemplo

incorporar datos de *flickr*, *YouTube* y *Google+*.

Además, *BDSP* soporta un número limitado de análisis, en que la *GUI* de *BDSP* maneja muy pocas opciones de configuración de los *frameworks* de análisis integrados. Sin embargo en estos *frameworks* ya se encuentran implementados una gran cantidad de análisis que pueden ser añadidos fácilmente a *BDSP*. Por ejemplo, *Mahout* cuenta una gran variedad de algoritmos especializados en diferentes áreas tales como filtros colaborativos y de reducción de dimencionalidad que no pudieron ser hechos accesibles a través de *BDSP* por falta de tiempo.

6.1.2. Mejora de elementos actuales

Una de las limitaciones que más impacta el desempeño de *BDSP* es el planificador de procesos que se utiliza, el cual se describe en el capítulo 4. Este planificador de procesos es muy sencillo y su funcionamiento consiste en agregar registros a una tabla en una base de datos *MySQL* con la cual se controlan los procesos de recuperación de archivos de datos y de análisis de datos pendientes por realizar. De esta manera, los datos se obtienen de manera secuencial en el mismo orden en que son especificados. Sería conveniente mejorar este planificador para permitir que distintos tipos de archivos se obtengan de manera concurrente

También se podría modificar el planificador de procesos para que particione el *cluster* de procesamiento y pueda ejecutar varios análisis al mismo tiempo, cada análisis utilizando distintos nodos, según la disponibilidad de recursos en el *cluster*. Con estos cambios se mejoría el *throughput* de las tareas de análisis.

Otro cambio posible consiste en modificar el planificador de procesos para permitir el procesamiento de datos en un *cluster Hadoop* privado administrado por el usuario. Esta característica podría ser de ayuda a usuarios de empresas o instituciones que cuenten con una gran cantidad de datos ya que podrían utilizar *BDSP* con su propio cluster de procesamiento. Opcionalmente se podría remplazar el cluster privado por servicios de procesamiento basados en la nube como *Elastic MapReduce* de *Amazon*.

6.1.3. Nuevas funcionalidades

Zepellin es una herramienta *Web* de visualización de datos que permite realizar gráficas de los datos del usuario. Se podría implementar *Zepellin* en *BDSP* y permitir al usuario generar gráficas de los datos almacenados y de los resultados de los análisis.

Implementar la plataforma *Spark* como complemento de *Hadoop* mejoraría enormemente el rendimiento de los análisis computacionalmente intensivos. Dependiendo del tipo de problema Spark puede ser hasta cien veces más rápido que *Hadoop* [36] ya que todas las operaciones son realizadas directamente sobre la memoria de la computadora, sin realizar ninguna escritura de datos intermedios al disco duro.

Sería interesante agregar un módulo que cuente con una interfaz donde el usuario pueda realizar consultas a sus datos utilizando lenguajes como *SQL*, *Scala* o R. En la versión actual de BDSP únicamente es posible visualizarlos y analizarlos pero no es posible realizar consultas en ningún lenguaje sobre estos datos.

La ausencia de programación visual es una carencia importante de *BDSP*. La programación visual permite especificar la obtención de datos y la realización de análisis de una manera muy intuitiva. Esta funcionalidad está presente en la todas las herramientas comerciales que se han descrito en el capítulo 2.

En resumen, esta tesis ha presentado un sistema *Web* que creemos es útil para facilitar el desarrollo proyectos *Big Data*. Puede servir como prototipo inicial de un proyecto para extenderlo según se requiera, y como vehículo de capacitación en análisis de datos y en desarrollo de *software Big Data*. Incluye ya el manejo de diferentes fuentes Web remotas, lo cual facilita los dos puntos anteriores.

Apéndice A

Reducción de dimensionalidad

Un problema central en el análisis de datos es la reducción del número de atributos de un conjunto de objetos (datos), es decir, intentar describir con precisión los valores de p variables utilizando un pequeño subconjunto $r < p$ de ellas. Encontrar este subconjunto r es lo que se conoce como análisis de reducción de dimensionalidad o ARD [68].

Los algoritmos de ARD implican la pérdida de una pequeña parte de la información original. Sin embargo son útiles para desechar información redundante y optimizar procesos de transformación o análisis posteriores.

El análisis de componentes principales es uno de los algoritmos más utilizados para realizar ARD. Es una técnica estadística que transforma un conjunto de datos con muchas variables en un nuevo conjunto de datos que contiene un menor número de variables diferentes pero que están correlacionadas con las variables iniciales.

En este apéndice se describe el análisis de componentes principales y se muestra en ejemplo del uso de este análisis en BDSP. Si se desea más información sobre otros métodos de reducción de dimensionalidad puede consultarse [68].

A.1. Análisis de componentes principales

El objetivo del análisis de componentes principales o PCA (del inglés *Principal Component Analysis*) es encontrar un nuevo conjunto de atributos que mejor capturen la variabilidad de los datos. Estos nuevos atributos son conocidos como dimensiones. En PCA la primera dimensión es elegida de tal manera que capture la mayor variabilidad posible. La segunda dimensión es ortogonal a la primera y captura la mayor variabilidad posible restante. El término ortogonal es una generalización de la noción geométrica de perpendicularidad.

PCA ayuda a identificar patrones en los datos, por lo que algunas veces es utilizado como una técnica para el reconocimiento de patrones [30]. Además, la mayor variabilidad de los datos es capturada por sólo una fracción de las dimensiones originales por lo que los resultados obtenidos con este análisis pueden ser fácilmente utilizados con cualquier otro tipo de análisis que no funcione adecuadamente con conjuntos de datos que tengan un alto número de dimensiones.

Detalles matemáticos

Supongamos que se representan los datos mediante una matriz D de m por n como se muestra a continuación:

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \cdots & d_{mn} \end{bmatrix}$$

Donde m son las filas que representan cada uno de los objetos y n son las columnas que representan los atributos de cada objeto. Se sugiere preprocesar la matriz D antes de iniciar el cálculo de PCA para facilitar el cálculo de la matriz de covarianza que a continuación se explica. Este preprocesamiento consiste en restar a cada atributo su media aritmética.

Se puede resumir la variabilidad de una colección de datos (objetos) con atributos continuos calculando una matriz de covarianzas S. La matriz de covarianza de la

matriz D es otra matriz S cuyos elementos s_{ij} se calculan de la siguiente manera:

$$s_{ij} = covarianza(d_{*i}, d_{*j}).$$

Esto significa que la covarianza del elemento s_{ij} es la covarianza del i-ésimo y del j-ésimo atributo (columna) de los datos. La covarianza de dos atributos se calcula de la siguiente manera:

$$Cov_{xy} = \sigma_{xy} = \frac{1}{n} \sum f_i (x_i - \overline{x})(y_i - \overline{y})$$

La covarianza es una medida que indica qué tanto varían los atributos entre sí. Si la matriz de datos D se ha preprocesado (restando a cada atributo su media aritmética) el promedio de cada atributo es 0, por lo que $S = D^T D$.

El objetivo de PCA es encontrar una transformación de los datos que satisfagan las siguientes propiedades [30]:

1. Cada nuevo par de atributos debe tener covarianza 0.

2. Los atributos deben estar ordenados con respecto a su varianza.

3. El primer atributo (dimensión) debe capturar la mayor variabilidad posible.

4. Los atributos siguientes deben ser ortogonales entre sí.

Esta transformación se puede obtener mediante el cálculo de los valores propios de la matriz de covarianza. Supongamos que $\lambda_1, \cdots, \lambda_n$ son los valores propios de S. Los valores propios de S son positivos por lo que pueden ser ordenados como sigue: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. A partir de estos valores podemos calcular una matriz de vectores propios $U = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$ de S. Recúerdese que estos vectores propios están ordenados por lo que el i-ésimo vector corresponde al i-ésimo valor propio más grande. Asumiendo que la matriz D se ha preprocesado se pueden decir las siguientes aseveraciones [30]:

- La matriz $D' = DU$ contiene un conjunto de transformaciones que satisfacen las condiciones en los siguientes incisos.
- Cada uno de los atributos de D' es una combinación lineal de los atributos originales. Una combinación lineal es una expresión matemática que consiste en la suma entre pares de atributos de dos o más conjuntos, multiplicados entre sí.
- La varianza del i-ésimo atributo de D' es λ_i.
- La suma de la varianza de los atributos originales es igual a la suma de la varianza de los nuevos atributos (los atributos de D' son los nuevos atributos).
- Los nuevos atributos son llamados componentes principales.

Los vectores propios de S definen un nuevo conjunto de índices. De tal manera que PCA puede ser visto como una rotación respecto a los índices originales. Estos nuevos índices estarán alineados con la variabilidad de los datos.

A.2. Ejemplo en BDSP

Para este ejemplo se aplicará el análisis de componentes principales al conjunto de datos Iris que se utilizó en el segundo ejemplo del capítulo 5. Este conjunto de datos es multivariable (multiatributo) y fue propuesto por Ronald Fisher. El conjunto cuenta con 50 muestras de tres tipos de flores iris (Iris setosa, Iris virginica e Iris versicolor). Cada una de estas muestras incluye cuatro medidas (atributos): el largo y ancho en centrímetos del sépalo y del pétalo.

El archivo de datos Iris datos consiste en un archivo en formato .CSV en donde el primer renglón es el encabezado que contiene el nombre de las columnas o atributos. A continuación se muestran los primeros cinco registros (renglones) del archivo:

sepal length,sepal width,petal length,petal width,class

5.1,3.5,1.4,0.2,Iris-setosa

4.9,3,1.4,0.2,Iris-setosa

4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa

Obtención de datos

Para mostrar que es posible realizar más de un análisis a un archivo de datos, en este ejemplo se asumirá que este archivo de datos fue agregado a BDSP previamente por el usuario tal y como se especifica en el capítulo 5.

Se recordará que en el capítulo 5 se realizó un análisis de agrupamiento utilizando K-means sobre el archivo de datos Iris. Para esto, el usuario tuvo que elegir un atributo para el eje X y otro atributo para el eje Y. En el ejemplo del capítulo 5 se eligieron los valores del ancho y largo del cépalo para el eje X y Y respectivamente. Esta elección se hizo porque se conocía de antemano que seleccionar estos dos atributos produce los mejores resultados con K-means y este conjunto de datos.

Si no se cuenta con esta información, se puede realizar un análisis de componentes principales, que generará un nuevo conjunto de datos en donde los primeros atributos (componentes) contienen la mayor variabilidad. Y la idea es, elegir los componentes primero y segundo para los ejes X y Y respectivamente, para el análisis de agrupamiento.

Análisis de datos

Para realizar el análisis de componentes principales utilizando BDSP, el usuario debe seleccionar en el menú *Analyses*/*Add Analysis* y seleccionar la opción *Dimensionality reduction*/PCA. Debe indicar el nombre del nuevo archivo de datos que se creará y el número k de componentes a obtener. Para este ejemplo se nombrará al nuevo archivo de datos como *Iris PCA K=4* y se establecerá k=4. Este formulario puede observarse en la figura A.1.

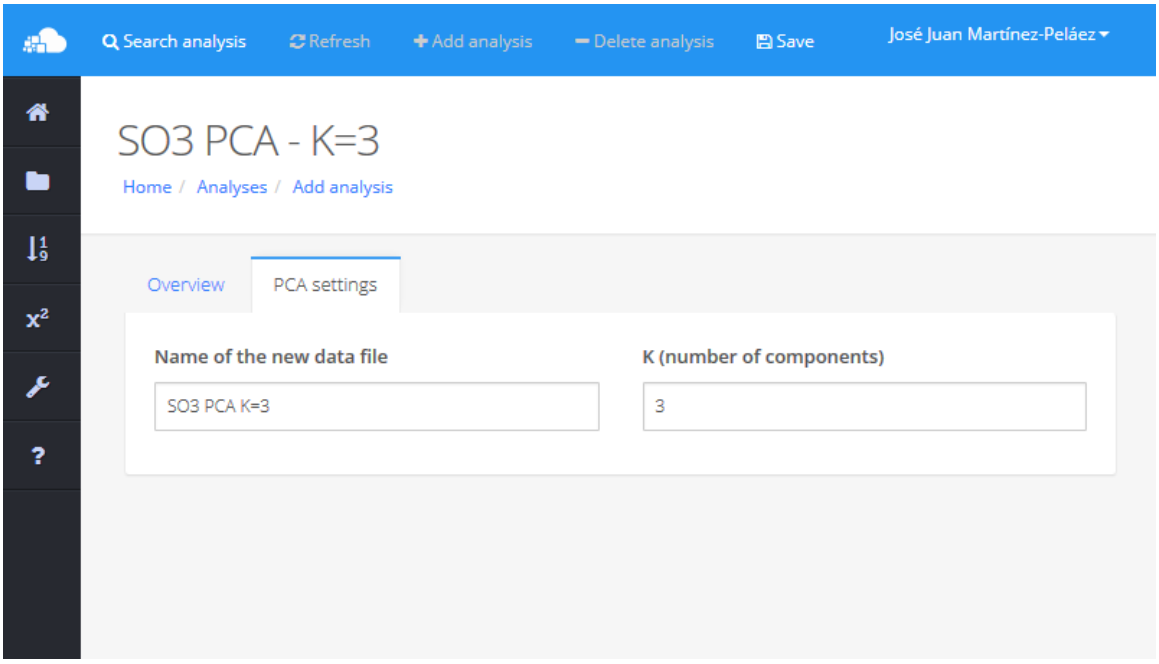

Figura A.1: Formulario de configuración del análisis PCA.

Resultados

Una vez finalizado el análisis se creará un nuevo archivo de datos en *BDSP* que puede ser seleccionado para realizar otros análisis tal como el análisis de agrupamiento. A continuación se muestran los primeros cinco registros (renglones) del archivo de datos generado:

c1,c2,c3,c4

-2.684125626,-0.319397247, 0.027914828, 0.0022624371

-2.714141687, 0.177001225, 0.210464272, 0.0990265503

-2.888990569, 0.144949426, -0.017900256, 0.0199683897

-2.745342856, 0.318298979, -0.031559374, -0.0755758166

-2.728716537, -0.326754513, -0.090079241, -0.0612585926

Si realizamos un análisis de agrupamiento a este archivo, tal y como se describe en el capítulo 5 obtendremos el resultado mostrado en la figura A.2.

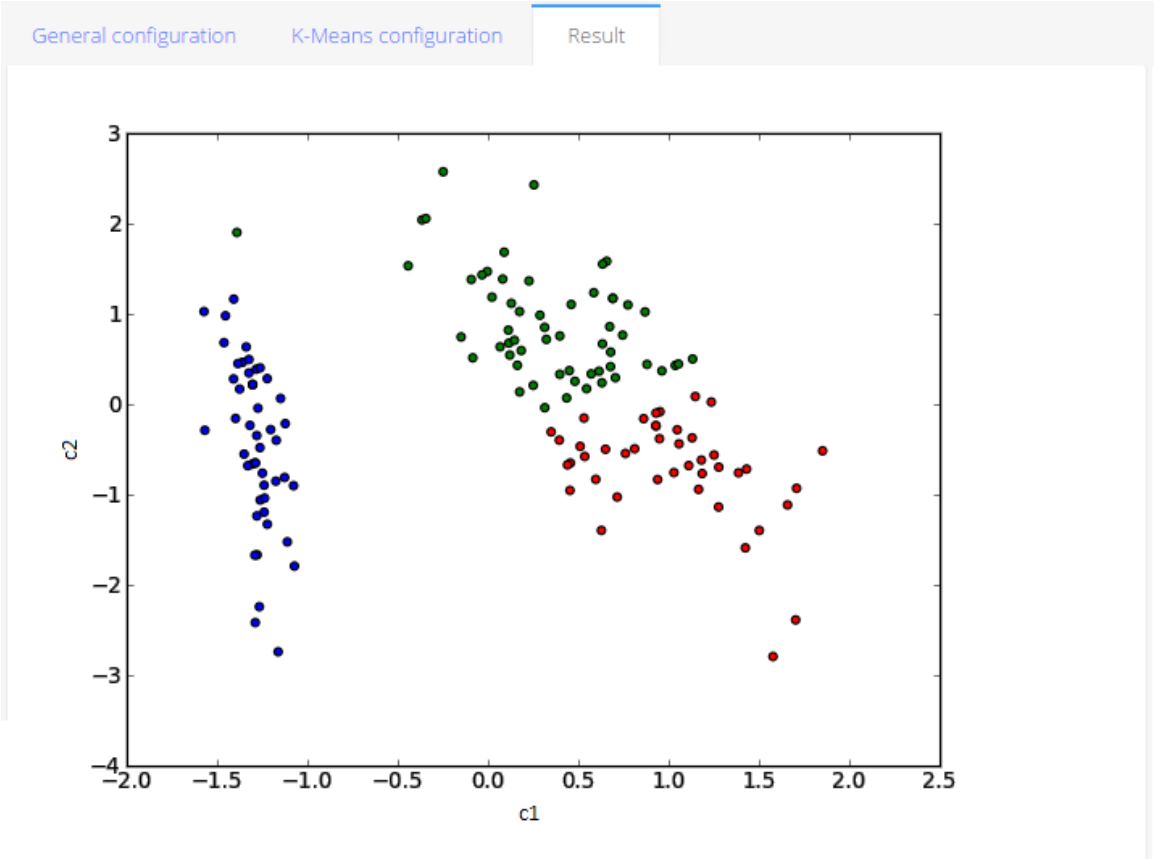

Figura A.2: Resultado del análisis *K-means* al conjunto de datos Iris con PCA.

A pesar que no se contaba con información previa del conjunto de datos Iris, fue posible obtener un resultado de agrupamiento con K-means similar al resultado del capítulo 5 que se observa en la figura A.3.

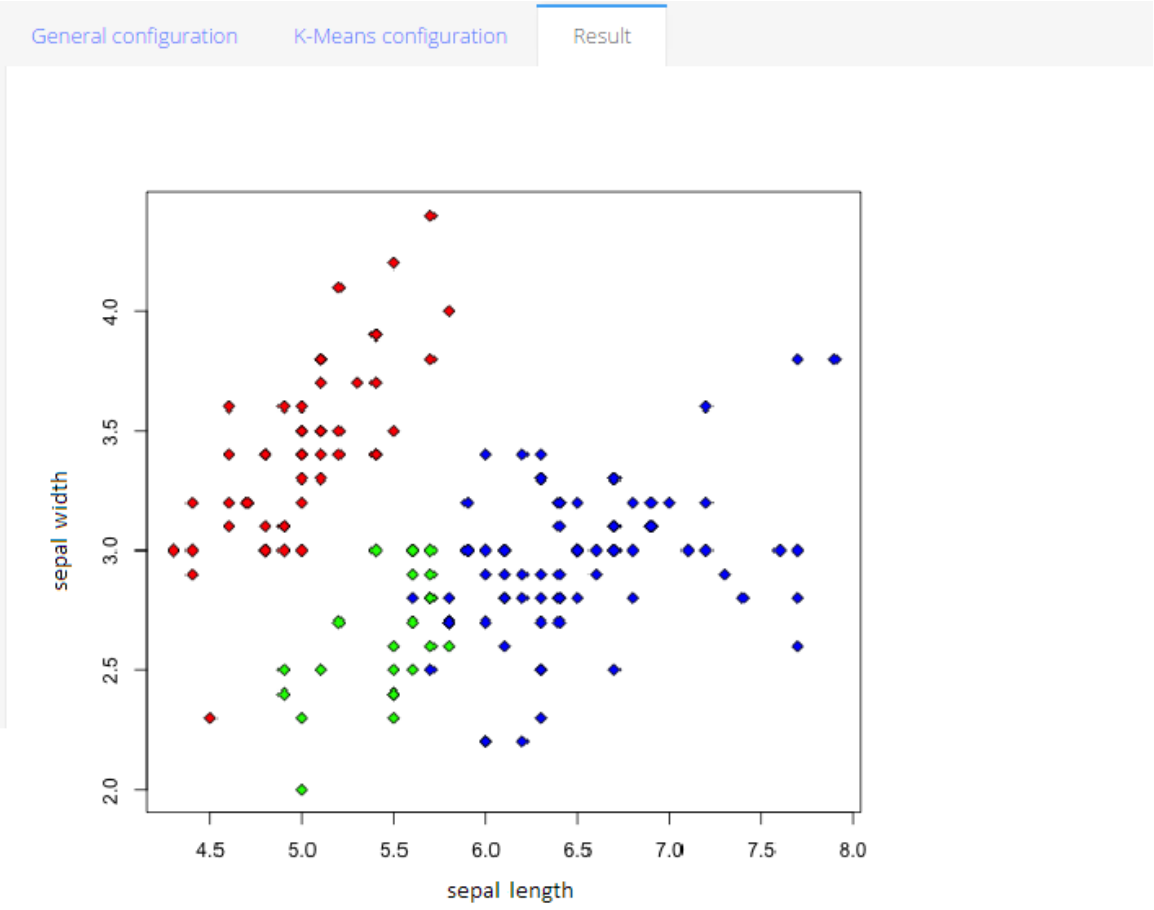

Figura A.3: Resultado del análisis *K-means* al conjunto de datos Iris.

Apéndice B

Instalación de BDSP

En este apéndice se detalla la instalación de *BDSP* en un servidor en el que únicamente se ha instalado el sistema operativo *Debian* 8. Es posible instalar *BDSP* en cualquier otra distribución de *Linux* haciendo algunos cambios a los comandos presentados en este apéndice.

B.1. Instalación del servidor *LAMP*

Antes de iniciar con la instalación de *BDSP* se debe preparar el servidor instalando *Apache HTTP*, *MySQL*, *PHP* y la extensión *PHP5-MySQL*; para esto se deben utilizar las siguientes instrucciones:

```
1 sudo -s
2 apt-get update
3 apt-get install apache2 mysql-server php5 php5-mysql
```

Todos los comandos mostrados requieren privilegios de *root*. La instrucción *sudo –s* iniciará un *Shell* con el usuario *root* y evitará la necesidad de agregar el comando sudo antes de cada instrucción sucesiva. Deberá proporcionar la contraseña del usuario *root* antes de continuar.

apt-get update se utiliza para actualizar la lista de paquetes de software disponibles en los repositorios configurados. El parámetro *update* no debe confundirse con

el parámetro *upgrade*, el primero (*update*) sólo actualiza la información de la lista de paquetes disponibles proporcionando la ruta donde se puede descargar la última versión disponible de todos los paquetes. El segundo parámetro (*upgrade*) sí realiza la actualización de los paquetes con la última versión conocida por la lista. El uso de *upgrade* es opcional aunque recomendado.

La tercer instrucción es la que instala los programas requeridos para ejecutar BDSP. Dependiendo de la configuración del servidor es posible que tenga que descargar algunos archivos de Internet. Deberá proporcionar la contraseña para el usuario root de MySQL cuando el sistema lo solicite. El comando apt-get instalará además de los paquetes especificados algunos otros que son necesarios.

B.2. Instalación de BDSP

El siguiente paso es copiar todos los archivos de BDSP a la ruta */var/www/bdsp*. Se debe especificar como propietario de estos archivos al usuario *www-data* y asignar el permiso 644 a todos los archivos y 755 a todos los directorios. Para realizar estas acciones se deben utilizar los siguientes comandos:

```
4 cd /var/www
5 cp -r /home/admin/bdsp/ .
6 chown -R www-data:www-data bdsp/
7 find bdsp/ -type f -exec chmod 644 {} \;
8 find bdsp/ -type d -exec chmod 755 {} \;
```

La cuarta instrucción nos posiciona dentro del directorio */var/www*, el directorio público por defecto de *Apache HTTP* en *Debian*, de tal manera que el directorio de *BDSP* será */var/www/bdsp*. La quinta instrucción copia todos los archivos de *BDSP*. En esta instrucción el usuario debe remplazar */home/admin/bdsp/* por la ruta en donde se encuentran los archivos de *BDSP*. La sexta instrucción asigna como propietario de los archivos de BDSP al usuario *www-data*, el usuario por defecto con el que se ejecutan las peticiones realizadas al servidor de Apache. Finalmente la séptima

```
root@ip-172-31-22-110:/var/www#
root@ip-172-31-22-110:/var/www# ls -l bdsp/
total 76
drwxr-xr-x  3 www-data www-data 4096 Sep 27 08:04 Capa1
drwxr-xr-x  2 www-data www-data 4096 Sep 27 08:04 Capa2
drwxr-xr-x  9 www-data www-data 4096 Sep 27 08:04 Capa3
-rw-r--r--  1 www-data www-data 1150 Sep 27 08:04 favicon.ico
drwxr-xr-x  9 www-data www-data 4096 Sep 27 08:04 fdw
-rw-r--r--  1 www-data www-data 1170 Sep 27 08:04 fdw_configuracion.php
-rw-r--r--  1 www-data www-data 2562 Sep 27 08:04 index.php
drwxr-xr-x  9 www-data www-data 4096 Sep 27 08:04 Mpsoft.FDW
drwxr-xr-x  3 www-data www-data 4096 Sep 27 08:04 nbproject
drwxr-xr-x 14 www-data www-data 4096 Sep 27 08:04 pagina
drwxr-xr-x  3 www-data www-data 4096 Sep 27 08:04 plantilla
drwxr-xr-x  3 www-data www-data 4096 Sep 27 08:04 _repositorio
-rw-r--r--  1 www-data www-data   26 Sep 27 08:14 robots.txt
drwxr-xr-x  2 www-data www-data 4096 Sep 27 08:04 _tmp
root@ip-172-31-22-110:/var/www#
```

Figura B.1: Estructura de directorios de BDSP

y octava instrucción asigna los permisos adecuados a cada uno de los archivos. La imagen B.1 muestra el resultado final de estas instrucciones.

B.3. Configuración del servidor Apache

B.3.1. Peticiones seguras

Ya que se han copiado los archivos de BDSP y se han asignado los permisos correspondientes se deberá configurar el servidor Apache HTTP para aceptar y procesar peticiones seguras HTTPS y para soportar la reescritura de URLs, para esto es necesario ejecutar las instrucciones que a continuación se presentan.

```
9  a2enmod ssl rewrite
10 service apache2 restart
```

Posteriormente se deberá crear el certificado de seguridad que será utilizado por Apache HTTP para procesar las peticiones HTTPS. Este certificado autogenerado

puede ser remplazado por un certificado emitido por una entidad certificadora de confianza. Los certificados autogenerados, como el propuesto en este apéndice, provocan advertencias de seguridad en todos los navegadores modernos. El certificado puede ser generado utilizando la siguiente instrucción:

```
11  openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /home/
        admin/bdsp.key -out /home/admin/bdsp.cer
```

La instrucción número 11 creará dos archivos en el directorio */home/admin*. Puede cambiar esta ruta pero se recomienda que el archivo *bdsp.key* se encuetre en una ruta segura no accesible públicamente, ya que es la llave privada del certificado generado.

B.3.2. Host virtual

El siguiente paso es configurar el host virtual en el servidor Apache HTTP que será utilizado por BDSP. Para este ejemplo se utilizará el host bdsp.com. Para esto se debe crear el archivo */etc/apache2/sites-available/bdsp.conf* con el siguiente contenido:

```
<VirtualHost *:443>
12          ServerName bdsp.com
13          ServerAlias www.bdsp.com
14          DocumentRoot /var/www/bdsp/
15
16          <Directory /var/www/bdsp/>
17              AllowOverride none
18
19              IndexIgnore *
20              Options -Indexes
21
22              Order allow,deny
23              Allow from all
24
25              RewriteEngine On
26
```

```
27            RewriteCond %{QUERY_STRING} base64_encode[^()]*\([^)]*\) [OR]
28            RewriteCond %{QUERY_STRING} (<|%3C)([^s]*s)+cript.*(>|%3E) [
                  NC,OR]
29            RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [
                  OR]
30            RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
31            RewriteRule .* index.php [F]
32
33            RewriteBase /
34            RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization
                  }]
35            RewriteCond %{REQUEST_URI} !^/index\.php
36            RewriteCond %{REQUEST_FILENAME} !-f
37            RewriteCond %{REQUEST_FILENAME} !-d
38            RewriteRule .* index.php [L]
39        </Directory>
40
41        SSLEngine on
42        SSLCertificateFile /home/admin/bdsp.cer
43        SSLCertificateKeyFile /home/admin/bdsp.key
44    </VirtualHost>
```

Finalmente, se debe reiniciar el servidor Apache HTTP y BDSP estará listo para ser utilizado en el servidor.

Apéndice C

Instalación de *Hadoop* y *Mahout*

En este apéndice se detalla la instalación de *Hadoop* y *Mahout* en un servidor en el que únicamente se ha instalado el sistema operativo *Debian* 8 de 64 bits. Es posible instalar *Hadoop* y *Mahout* en cualquier otra distribución de Linux haciendo algunos cambios a los comandos presentados en este apéndice.

Antes de iniciar con la instalación de Hadoop es necesario instalar la máquina virtual de *Java*. *Mahout* es una herramienta que debe ser compilada antes de su uso por lo que además es necesario instalar Maven y Subversion. *Maven* es una herramienta de software para la gestión y construcción de proyectos. *Subversion* es una herramienta de control de versiones y permite descargar el código de proyectos que se encuentren en servidores que han implementado esta aplicación.

A continuación se muestra el procesedimiento para instalar los paquetes de *software* necesarios para el funcionamiento de *Hadoop* y *Mahout*. Posteriormente se explicará cómo instalar *Hadoop* y *Mahout*.

C.1. Instalación de *Java*

Java JDK es un conjunto de herramientas y bibliotecas que permiten compilar, ejecutar y generar documentación de programas en lenguaje *Java*. El paquete *Java JDK* incluye el ambiente de ejecución de Java necesario para utilizar *Hadoop* y

Mahout. Los comandos que instalan *Java JDK* son los siguientes:

```
1  sudo -s
2
3  sudo apt-get install python-software-properties
4
5  sudo apt-get install sun-java6-jdk
6
7  update-java-alternatives -s java-6-sun
```

La instrucción *sudo –s* se debe utilizar si el usuario con el que se está instalando BDSP no es el usuario root. Esta instrucción evitará la necesidad de agregar el comando sudo antes de cada instrucción sucesiva ya que iniciará un Shell con el usuario root. Deberá proporcionar la contraseña del usuario root antes de continuar.

C.2. Instalación de *Maven*

Maven es una herramienta de software para la gestión y construcción de proyectos en el lenguaje *Java*. *Maven* es útil para describir proyectos de *software* que se deben compilar. Almacena información sobre sus dependencias entre otros módulos y componentes externos, y proporciona el orden de compilación de los elementos.

A continuación se muestra el comando con el que se puede instalar *Maven*:

```
1  apt-get install maven
```

C.3. Instalación de *Subversion*

Subversion es una herramienta de control de versiones cuyo funcionamiento se asemeja al de un sistema de archivos. Puede acceder al repositorios a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras.

Se instala así:

```
1  apt-get install subversion
```

C.4. Instalación de *Hadoop*

El procedimiento para instalar Hadoop en un sólo nodo es muy sencillo. Sólo se debe descargar el archivo correspondiente a la versión de Linux que se esté utilizando y descomprimirlo. Hadoop está listo para ser utilizado en un sólo nodo. El archivo de instalación puede ser descargado desde:
http://www.apache.org/dyn/closer.cgi/hadoop/core

No obstante, bajo ciertas configuraciones pueden ocurrir problemas relacionados con permisos y rutas por lo que se aconseja seguir el siguiente procedimiento: Descomprimir el archivo de instalación de *Hadoop* en el directorio /usr/local/hadoop; Crear un usuario especial para la ejecución de Hadoop y asignarle todos los permisos para realizar cambios sobre los archivos de *Hadoop*, con los siguientes comandos:

```
1  sudo tar xzf hadoop-2.7.1.tar.gz
2  sudo mv hadoop-2.7.1 hadoop
3
4  addgroup hadoop
5  adduser --ingroup hadoop hduser
6  chown -R hduser:hadoop hadoop
```

Se debe modificar el contenido del archivo .bashrc, que se encuentra en el directorio home del usuario hduser con el fin de indicar la ruta en donde se ha instalado *Hadoop* y *Java*. El archivo debe tener el siguiente contenido:

```
1  # Directorio raíz de Hadoop
2  export HADOOP_HOME=/usr/local/hadoop
3
4  # Directorio raíz de Java
5  export JAVA_HOME=/usr/lib/jvm/java-6-sun
6
7  # Por conveniencia se agregan los siguientes alias
8  unalias fs &> /dev/null
9  alias fs="hadoop fs"
10 unalias hls &> /dev/null
11 alias hls="fs -ls"
```

```
12
13  # Se añade el directorio bin/ de Hadoop a la variable PATH
14  export PATH=$PATH:$HADOOP_HOME/bin
```

Los comandos anteriores crean *alias* opcionales que facilitan la creación de *scripts* para realizar análisis de manera automática. *BDSP* no utiliza ninguno de estos *scripts* pero una gran cantidad de ejemplos disponibles en Internet sí los utilizan por lo que es recomendable tenerlos.

C.5. Instalación de Mahout

La instalación de Mahout es lenta y requiere de una gran cantidad recursos del sistema porque se debe descargar el código fuente de internet y compilarlo para poder utilizarlo. Dependiendo de la velocidad de conexión a Internet y las características del equipo de cómputo que se esté instalando todo este proceso puede tardar tres horas aproximadamente. Los comandos para realizar este procedimiento son los siguientes:

```
1  cd /home/mahout
2  svn co http://svn.apache.org/repos/asf/mahout/trunk
3  cd trunk
4  mvn install
```

El comando de la línea 2 descargará el código fuente de Mahout y lo colocará en el directorio actual. Este código está diseñado para ser construido utilizando Maven. El comando de la línea 4 realizará el compilado del código.

Bibliografía

[1] Maryam Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics.

[2] European Commision. Digital agenda for europe: Communication on data-driven economy. http://ec.europa.eu/digital-agenda/en/news/communication-data-driven-economy, 2015. Consultado en septiembre de 2015.

[3] A. Silberschatz, H. Korth, and S. Sudarshan. *Database Systems Concepts*. McGraw-Hill, Inc., New York, NY, USA, 6 edition, 2011.

[4] Johannes L. *Data Mining and Business Analytics with R*. John Wiley & Sons, 1 edition, 2013.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *6th Symposium on Operating System Design and Implementation (OSDI 2004), San Francisco, California, USA, December 6-8, 2004*, pages 137–150, 2004.

[7] David A. *The Google Story*. Pan, 2006.

[8] Wikipedia. Datallegro. http://en.wikipedia.org/wiki/DATAllegro, 2015. Consultado en septiembre de 2015.

[9] Wikipedia. Netezza. http://en.wikipedia.org/wiki/Netezza, 2015. Consultado en septiembre de 2015.

[10] Wikipedia. Greenplum. http://en.wikipedia.org/wiki/Greenplum, 2015. Consultado en septiembre de 2015.

[11] NLTK. Natural language toolkit. http://www.nltk.org, 2015. Consultado en septiembre de 2015.

[12] Apache.org. Mahout. scalable machine learning and data mining. http://mahout.apache.org, 2015. Consultado en octubre de 2015.

[13] Apache.org. Welcome to apache hadoop. https://hadoop.apache.org, 2015. Consultado en octubre de 2015.

[14] Apache. Apache zeppelin (incubating). https://zeppelin.incubator.apache.org, 2015. Consultado en octubre de 2015.

[15] Databricks. Databriks: Data science made easy, from ingest to production. https://databricks.com/product/databricks, 2015. Consultado en noviembre de 2015.

[16] RapidMiner. Rapid miner website. https://rapidminer.com, 2015. Consultado en septiembre de 2015.

[17] Pentaho. Pentaho: Business analytics and business intelligence. http://www.pentaho.com, 2015. Consultado en noviembre de 2015.

[18] KNIME. Knime website. https://www.knime.org, 2015. Consultado en octubre de 2015.

[19] IBM. Ibm spss software. http://www-01.ibm.com/software/analytics/spss/, 2015. Consultado en noviembre de 2015.

[20] IDC. The 2011 idc digital universe study. Technical Report White paper, International Data Corporation, 2011.

[21] IDC. The digital universe of opportunities: Rich data and the increasing value of the internet of things. Technical Report White paper, International Data Corporation, 2014.

[22] Douglas Laney. 3D data management: Controlling data volume, velocity, and variety. Technical report, META Group, February 2001.

[23] Office of Science and Technology Policy. Big data research and development plan. https://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release.pdf, 2012. Consultado en octubre de 2015.

[24] National Institutes of Health. Big data to knowledge (bd2k). https://datascience.nih.gov/bd2k, 2015. Consultado en octubre de 2015.

[25] Alexandros Labrinidis and H. V. Jagadish. Challenges and opportunities with big data. *Proc. VLDB Endow.*, 5(12):2032–2033, August 2012.

[26] S. Mohanty, M. Jagadeesh, and H. Srivatsa. *Big Data Imperatives: Enterprise Big Data Warehouse, BI Implementations and Analytics*. Apress, Berkely, CA, USA, 1st edition, 2013.

[27] Saso Dzeroski. Relational data mining. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pages 887–911. 2010.

[28] R.E. Walpole, R.H. Myers, and S.L. Myers. *Probabilidad y estadística para ingenieros*. Pearson Educación, 1999.

[29] R Doll, R Peto, K Wheatley, R Gray, and I Sutherland. Mortality in relation to smoking: 40 years' observations on male british doctors. *BMJ*, 309(6959):901–911, 1994.

[30] Vipin K. Pang-Ning T., Michael S. *Introduction to Data Mining*. Addison-Wesley, 2006.

[31] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, October 2004.

[32] Jian-Xin Pan, Wing-Kam Fung, and Kai-Tai Fang. Multiple outlier detection in multivariate data using projection pursuit techniques. *Journal of Statistical Planning and Inference*, 83(1):153 – 167, 2000.

[33] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[34] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[35] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2009.

[36] Apache. Sparck: Lightning-fast cluster computing. https://spark.apache.org, 2015. Consultado en octubre de 2015.

[37] Apache.org. Mahout 0.10.1 features by engine. https://mahout.apache.org/users/basics/algorithms.html, 2015. Consultado en octubre de 2015.

[38] KDnuggets. Kdnuggets 14th annual analytics, data mining, data science software poll. http://www.kdnuggets.com/2013/06/kdnuggets-annual-software-poll-rapidminer-r-vie-for-first-place.html, 2013. Consultado en octubre de 2015.

[39] KDnuggets. Kdnuggets 15th annual analytics, data mining, data science software poll. http://www.kdnuggets.com/2014/06/kdnuggets-annual-software-poll-rapidminer-continues-lead.html, 2014. Consultado en octubre de 2015.

[40] Mark van Rijmenam. Walmart is making big data part of its dna. https://datafloq.com/read/walmart-making-big-data-part-dna/509, 2013. Consultado en octubre de 2015.

[41] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute, June 2011.

[42] Wikipedia. Walmart. http://es.wikipedia.org/wiki/Walmart, 2015. Consultado en noviembre de 2015.

[43] SAS. Big data meets big data analytics. Technical Report White paper, SAS.

[44] Wikipedia. Business intelligence. https://en.wikipedia.org/wiki/Business_intelligence, 2015. Consultado en noviembre de 2015.

[45] Su-Yeon Kim, Tae-Soo Jung, Eui-Ho Suh, and Hyun-Seok Hwang. Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert Systems with Applications*, 31(1):101 – 107, 2006.

[46] David R. Bell, Jeongwen Chiang, and V. Padmanabhan. The decomposition of promotional response: An empirical generalization. *Marketing Science*, 18(4):504–526, 1999.

[47] Yi Cai, Raymond Y.K. Lau, Stephen S.Y. Liao, Chunping Li, Ho-Fung Leung, and Louis C.K. Ma. Object typicality for effective web of things recommendations. *Decision Support Systems*, 63:52 – 63, 2014.

[48] M. Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo J.G. Lisboa. The value of personalised recommender systems to e-business: A case study. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 291–294, New York, NY, USA, 2008. ACM.

[49] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[50] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[51] Kevin D. *The $1,000 Genome: The Revolution in DNA Sequencing and the New Era of Personalized Medicine*. Free Press, 1 edition, 2010.

[52] Michael C. Schatz. Cloudburst: highly sensitive read mapping with mapreduce. *Bioinformatics*, 25(11):1363–1369, 2009.

[53] Aisling ODriscoll, Jurate Daugelaite, and Roy D. Sleator. 'Big data', Hadoop and cloud computing in genomics. *Journal of Biomedical Informatics*, 46(5):774–781, October 2013.

[54] Dell. Children's cancer care. http://www.dell.com/learn/us/en/uscorp1/childrens-cancer-care, 2012. Consultado en noviembre de 2015.

[55] GenomeWeb. Nextbio, intel to collaborate on improving hadoop stack for genomic data analysis. https://www.genomeweb.com/informatics/nextbio-intel-collaborate-improving-hadoop-stack-genomic-data-analysis, 2012. Consultado en noviembre de 2015.

[56] MarketWired. Cloudera chief scientist jeff hammerbacher teams with mount sinai school of medicine to solve medical challenges using big data. http://www.marketwired.com/press-release/cloudera-chief-scientist-jeff-hammerbacher-teams-with-mount-sinai-school-medicine-1htm, 2012. Consultado en noviembre de 2015.

[57] Chen Xu and Chaowei Yang. Introduction to big geospatial data research. *Annals of GIS*, 20(4):227–232, 2014.

[58] Adriana Maria Wilhelmina. *Analysis of urban traffic patterns using clustering*. PhD thesis, Enschede, April 2007.

[59] Google. Documentos de google. https://www.google.com.mx/intl/es-419/docs/about/, 2015. Consultado en octubre de 2015.

[60] Adobe. Adobe creative cloud. http://www.adobe.com/mx/creativecloud.html, 2015. Consultado en noviembre de 2015.

[61] Computer Hope. Linux and unix crontab command. http://www.computerhope.com/unix/ucrontab.htm, 2013. Consultado en noviembre de 2015.

[62] Jasmina Smailovic, Miha Grcar, Nada Lavrac, and Martin Znidarsic. Stream-based active learning for sentiment analysis in the financial domain. *Information Sciences*, 296(1):181–201, 2014.

[63] jQuery. The jquery plugin registry. https://plugins.jquery.com/, 2015. Consultado en diciembre de 2015.

[64] Startbootstrap. Bootstrap resources. http://startbootstrap.com/bootstrap-resources/, 2015. Consultado en diciembre de 2015.

[65] Google. How one-time passwords work and how they integrate with http authentication. https://code.google.com/p/mod-authn-otp/wiki/OneTimePasswords, 2009. Consultado en diciembre de 2015.

[66] Yuen D., Dzwinel W., Ben zion Y., and Kadlec B. Earthquake clusters over multi-dimensional space, visualization of. pages 2347–2371, 2009.

[67] Wikipedia. Iris flower data set. http://en.wikipedia.org/wiki/Iris_flower_data_set, 2015. Consultado en septiembre de 2015.

[68] Imola Fodor. A survey of dimension reduction techniques. Technical report, 2002.