APLICACIÓN DE LA METODOLOGÍA CRISP-DM® A LA RECOLECCIÓN Y ANÁLISIS DE DATOS GEORREFERENCIADOS DESDE TWITTER®

GUSTAVO ADOLFO GARCÍA VÉLEZ

Proyecto de Grado presentado como requisito para optar al Título de

ESPECIALISTA EN GEOMÁTICA

UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN GEOMÁTICA
BOGOTÁ D.C., COLOMBIA
DICIEMBRE DE 2018

APLICACIÓN DE LA METODOLOGÍA CRISP-DM® A LA RECOLECCIÓN Y ANÁLISIS DE DATOS GEORREFERENCIADOS DESDE TWITTER®

APPLICATION OF THE CRISP-DM® METHODOLOGY TO RECOLLECTION AND ANALYSIS OF GEOREFERENCED DATA FROM TWITTER®

Gustavo Adolfo García Vélez, Ingeniero Ambiental, aspirante a Especialista en Geomática
Universidad Militar Nueva Granada, Bogotá, Colombia
Correo electrónico: u3101426@unimilitar.edu.co, ga.garcia57@uniandes.edu.co

Resumen– *La minería de datos es actualmente una de las áreas con mayor auge y éxito dentro de la informática, al permitir encontrar correlaciones y patrones a partir del análisis de grandes volúmenes de datos. En este sentido, la Geomática es fundamental si los datos se encuentran georreferenciados, aportando el componente espacial del análisis. Una metodología ampliamente utilizada en el desarrollo de proyectos de minería de datos es la denominada CRISP-DM®, compuesta de seis etapas (comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implementación), la cual se emplea para el análisis de información georreferenciada proveniente de la red social Twitter®, con el fin de hallar patrones que permitan responder preguntas como: ¿en dónde se generan más trinos geolocalizados en la ciudad de Bogotá? ¿Cuáles son los sectores catastrales de Bogotá en donde sería más probable encontrar un tweet georreferenciado?*

Palabras Clave– Análisis de Puntos Calientes, API, CRISP-DM, Densidad Kernel, Minería de Datos, Python, PostgreSQL, Twitter

Abstract– *Data mining is currently one of the most successful areas in informatics since it allows finding correlations and patterns from analysis of big data. In that sense, Geomatics is fundamental as long as data is georeferenced, by giving the spatial component of the analysis. CRISP-DM® is a widely-used methodology in data mining with six basic steps: business understanding, data understanding, data preparation, modeling, evaluation and deployment, which will be used for analysis of geolocated data from Twitter® social network in order to find patterns and answer questions such as: In what places of Bogotá more georreferenced tweets are generated? In which cadastral sectors is likely to find a located tweet?*

Key words– API, CRISP-DM, Data Mining, Hot Spot Analysis, Kernel Density, Python, PostgreSQL, Twitter

1. Introducción

La generación de datos en nuestro mundo tecnológico es permanente y va en constante aumento: en un segundo se envían cerca de 8500 trinos (mensajes en la red social Twitter), se suben 880 fotografías en Instagram, y se genera un tráfico de 65 GB de datos a través de todo Internet. Tales volúmenes de datos requieren técnicas especiales de procesamiento y análisis para su conversión en *información útil* para un propósito (negocios, investigación, visualización, etc.). Es en este sentido que surge la *minería de datos* (*data mining*) como una de las ramas del análisis creada con el fin de extraer *información implícita u oculta, previamente desconocida y potencialmente útil* de dichos grandes volúmenes [1]. La minería de datos implica la conversión de un conjunto bruto de *datos* (hechos individuales) en *información* (datos relacionados) y en *conocimiento* (patrones) [2].

Colombia es una fuente muy importante de datos, teniendo en cuenta que el 86% de los hogares del país tiene algún tipo de acceso a Internet y se estima que redes sociales como Facebook tienen más de 15 millones de usuarios activos, superando a países europeos como Francia y Alemania [3].

Twitter, la red social de interacción a través de mensajes de texto y contenido multimedia, es utilizada por aproximadamente 6 millones de usuarios en Colombia, y constituye un medio oficial de expresión para entidades gubernamentales, empresas y personas a través de *trinos* o *tweets* [3].

Desde el punto de vista de la *geomática*, cabría preguntarse qué proporción de datos generados en el país tiene algún tipo de ubicación espacial (está georreferenciada) y cuál es el propósito y contenido de estos datos, en el caso de estar disponibles para uso público.

Este trabajo tiene como objetivo principal la aplicación de una metodología de minería de datos denominada *CRISP-DM*® sobre trinos o *tweets* provenientes de Twitter, empleando una aplicación escrita en lenguaje *Python*, capaz de recolectar los datos, clasificarlos, depurarlos y almacenarlos en una base de datos espacial, con el fin de generar conocimiento al restringir la búsqueda a un espacio geográfico definido (Bogotá) y colectando únicamente los datos georreferenciados para determinar patrones de agrupamiento espacial en la ciudad y comprender mejor este fenómeno.

Es preciso anotar que el autor cuenta con la debida autorización de Twitter para el uso de los datos recolectados, a través de una cuenta de desarrollador en la API (*Application Programming Interface* – Interfaz de programación de Aplicaciones) de esta red de *microblogging*.

2. Marco Teórico

2.1 Minería de Datos

La minería de datos, adaptación del término *data mining* en inglés, se define como el proceso de *extracción* de información *implícita* u *oculta*, no plenamente explotada y en la mayoría de los casos no conocida previamente, a partir de datos disponibles en repositorios (bases de datos) o en tiempo real. Con este proceso se pretende *generar conocimiento útil* a partir del establecimiento de *relaciones entre los datos*, tales como patrones o asociaciones [2].

En síntesis, los procesos de minería de datos transforman *grandes conjuntos de datos* en *conocimiento* que sea de utilidad para la toma de decisiones o para comprender mejor la realidad o un fenómeno concreto [2].

Teniendo en cuenta la definición anterior, el término "minería de datos" es equiparable a expresiones como *extracción de conocimiento, descubrimiento de conocimiento en repositorios y bases de datos, análisis de datos y obtención de patrones* [1].

Es necesario aclarar que la minería de datos no es una simple *búsqueda* de datos concretos, empleándose para ello *consultas* sobre la base de datos respectiva. Para ciertos conjuntos de datos, la estadística descriptiva o las consultas no son suficientes para su análisis, siendo necesario emplear los algoritmos de descubrimiento de conocimiento (minería de datos propiamente dicha).

A continuación, se define la metodología para realizar un proceso efectivo de minería de datos, con la descripción de sus respectivas etapas.

2.2 La metodología CRISP® para desarrollo de proyectos en minería de datos

No existe una única metodología o procedimiento para el desarrollo de un proyecto de minería de datos. Gallardo [4] y León [3], citan las siguientes:

SEMMA (*Sample, Explore, Modify, Model, Assess*), desarrollado por SAS Institute Inc. en 2003.

KDD – *Knowledge Discovery in Databases* (*Selección, Preprocesamiento y limpieza, Transformación, Minería de Datos, Evaluación e Implantación*), propuesto por Fayyad en 1996.

CRISP-DM – *CRoss Industry Standard Process for Data Mining*, financiado en 1999 por la entonces Comunidad Económica Europea (CEE), hoy Unión Europea (UE).

CRISP-DM es la guía de referencia en desarrollo de proyectos de minería de datos más ampliamente utilizada en el mundo, al surgir como una iniciativa de varias empresas privadas (*NCR, AG, SPSS, Daimler-Chrysler*), que en el marco de la CEE idearon una guía de referencia de *libre distribución* basada en diferentes versiones de la metodología KDD [4].

Entre las ventajas de utilizar CRISP-DM, se destaca la posibilidad de replicación de proyectos, su independencia de la industria, aplicación o proyecto; su neutralidad con respecto a las herramientas y su enfoque en las situaciones de negocios y en el análisis técnico. En últimas, ayuda al proceso de planeación y gerencia del proyecto de minería de datos [2].

La metodología CRISP-DM® se resume en el modelo de proceso que se ilustra en la figura 1.

Las etapas en las que se divide CRISP® mencionadas por Gallardo [4] se describen a continuación.

2.2.1 Comprensión del negocio

Esta etapa implica el conocimiento y entendimiento de los propósitos y requerimientos del negocio desde la perspectiva empresarial o académica, con el objetivo de transformarlos en metas de tipo técnico y desarrollar un plan de proyecto de minería de datos.

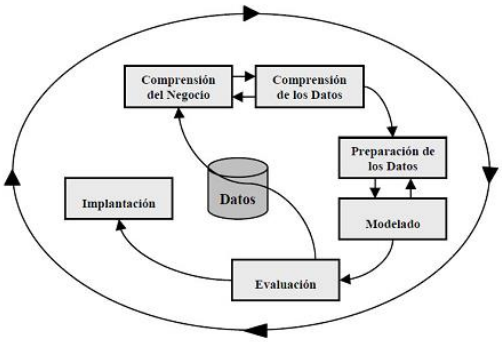

Figura 1. Flujo de proceso de la metodología CRISP® (*Fuente*: [4])

En este primer nivel es fundamental el entendimiento completo del problema que desea resolverse, lo cual permitirá recolectar los datos necesarios y la interpretación coherente de los resultados de los análisis.

La comprensión del negocio se divide en las siguientes cuatro tareas o subprocesos: *Determinación de los objetivos del negocio, Valoración de la situación, Determinación de los objetivos de la minería de datos* y *Generación de un plan de proyecto* [4].

2.2.2 Comprensión de los datos

Posterior a la comprensión del negocio, es necesario realizar una recolección inicial de datos con el fin de familiarizarse con ellos, verificando atributos como su calidad y estableciendo las relaciones más evidentes entre ellos. Esta etapa se conoce como la *comprensión de los datos*, y junto con la comprensión del negocio implican el mayor tiempo y esfuerzo en un proyecto de minería de datos.

La comprensión de los datos se divide en cuatro tareas, a saber: *Recolección de datos iniciales, Descripción de los datos, Exploración de los datos, Verificación de la calidad de los datos* [4].

2.2.3 Preparación de los datos

La fase de preparación de datos agrupa los procesos de selección, limpieza, generación de variables o atributos adicionales, integración de diversos orígenes de datos y cambios de formato. Lo anterior garantiza la adaptación de los datos recolectados a los procesos y algoritmos de minería de datos que se empleen posteriormente.

Lo anterior necesariamente implica que esta fase de preparación se encuentra estrechamente relacionada con la siguiente fase de modelado, por cuanto es diferente según el modelo o algoritmo a utilizar.

Los subprocesos de esta fase son: *Selección de datos, Limpieza de los datos, Estructuración de los datos, Integración de los datos***,** *Formateo de los datos* [4].

2.2.4 Modelado

En esta fase se selecciona la técnica de modelado más apropiada para el proyecto de *data mining*, teniendo en cuenta los siguientes criterios:

- La técnica debe ser *apropiada para el problema*.
- La técnica debe disponer de los *datos adecuados* según el propósito del proyecto.
- La técnica debe *cumplir los requisitos del problema*, esto es: dar solución a los requerimientos planteados en el plan de proyecto.
- La técnica debe tener un *tiempo adecuado para obtener un modelo.* El "tiempo adecuado" depende de los requerimientos del proyecto, y se espera que no supere cierto umbral.
- Conocimiento de la técnica.

La fase de modelado implica las siguientes etapas genéricas: *Selección de la técnica de modelado, Generación del plan de prueba, Construcción del modelo* y *Evaluación del modelo* [4].

2.2.5 Evaluación

En esta etapa de la metodología se evalúa el modelo escogido en la fase anterior, considerando los siguientes aspectos:

- Criterios de éxito del problema. ¿Se obtuvo una solución para el problema planteado en el plan de proyecto?
- ¿La fiabilidad de los resultados es aplicable a todo el espectro de datos, teniendo en cuenta que solo se empleó una muestra de los mismos?
- ¿Qué herramientas se pueden emplear para la evaluación de los resultados?

La etapa de evaluación involucra los siguientes subprocesos: *Evaluación de resultados*, *Revisión* y *Determinación de próximos pasos* [4].

2.2.6 Implementación

A esta fase se llega una vez que el modelo de minería de datos ha sido validado. Consiste en la transformación del *conocimiento* obtenido en acciones dentro del proceso de negocio, a través de dos mecanismos, a saber:

- Recomendaciones de un intérprete de los resultados o *analista*, para tomar acciones basadas en los resultados del modelo.

- Aplicación del modelo en otros conjuntos de datos y su incorporación al modelo de negocio.

El modelo así obtenido debe documentarse adecuadamente para lograr su entendimiento por parte de todos los usuarios y de esta manera incrementar el conocimiento.

La implementación consta de cuatro subprocesos: *Plan de implementación*, *Monitorización y Mantenimiento*, *Informe Final* y *Revisión del proyecto* [4].

Una vez descrita la metodología CRISP-DM®, así como sus etapas principales, es necesario definir el negocio y los datos a procesar, las *herramientas de trabajo* y las técnicas de modelado para satisfacer los requerimientos de la metodología, lo cual se trata a continuación.

2.3 El servicio de *microblogging* de Twitter. El API de Twitter

Twitter es un servicio de *microblogueo* (*microblogging* en inglés) para compartir información (texto, imágenes, videos, etc.) de una forma rápida y sencilla. Entra en la categoría más amplia de ***red social*** ya que permite la interacción entre usuarios. Creada por Jack Dorsey en 2006, se basa en el envío de mensajes cortos (denominados *tweets* en inglés, *trinos* en español) de 140 caracteres, ampliados a 280 caracteres a partir de noviembre de 2017 [5].

En Colombia, Twitter es un medio de prestigio, en donde las organizaciones gubernamentales, empresas privadas, personajes de la vida pública y de la farándula realizan anuncios y dan a conocer sus opiniones, al igual que millones de personas más. Los trinos frecuentemente son fuente de controversia y de debate. Se estima que existen cerca de 6 millones de usuarios activos de Twitter en el país.

Desde el punto de vista de la minería de datos, Twitter es una de las "vetas" más sobresalientes, siendo la fuente de numerosas investigaciones y estudios, tanto por la variedad como por el gran volumen de información generado.

Twitter ha dispuesto una API (*Application Programming Interface* – Interfaz de Programación de Aplicaciones), un conjunto de instrucciones, comandos y protocolos que pueden ser empleados por otro *software* para acceder a diversos servicios (extracción de información, modificación de mensajes, actualizaciones de estado, etc.) [6].

El acceso a la API de Twitter requiere un *punto de conexión*, una dirección que corresponde a una forma específica de información que proporciona el sistema. De esta forma, es posible crear aplicaciones que se integren con Twitter, las cuales deben *registrarse* ante la organización de *microblogging*.

Existen cinco tipos principales de puntos de conexión, a saber:

- Cuentas y usuarios
- Trinos (*tweets*) y respuestas
- Mensajes directos
- Anuncios
- Herramientas SDK y del editor

Para el propósito de este trabajo, interesan los *trinos y respuestas*, al ser información directamente generada por los usuarios y de uso público.

Un *tweet* no es solo un texto plano o un vínculo a otro sitio de Internet. Es un objeto complejo, definido por varios atributos que se relacionan con su respectivo valor en una estructura de empaquetamiento llamada *JSON* (*JavaScript Object Notation*) [7].

Un *tweet object* u "Objeto trino" es el tipo de dato definido en Twitter para el trino de un usuario. Consta de atributos *raíz* y atributos *extendidos*. Entre los atributos raíz se destacan los siguientes [8]:

`created_at` atributo que indica la fecha de creación del *tweet.*

`text`, atributo que almacena el texto original del *tweet* publicado por un usuario.

`source`, servicio o aplicación empleada por el usuario para publicar el *tweet*.

`coordinates`, atributo que representa las coordenadas geográficas en el sistema EPSG 4326 (WGS84) del *tweet* como las reporta el usuario a través de un determinado servicio. En sí mismo es otro objeto de tipo *geoJSON*, con la localización representada por

una *longitud* y una *latitud*. Este atributo puede tomar valores nulos (el usuario puede optar por no georreferenciar sus *tweets*).

`place`, atributo que indica que el *tweet* está asociado a un lugar determinado, pero no siempre se origina desde allí. *Place* es también un objeto complejo, definido por unos límites georreferenciados (*bounding box*).

`lang`, atributo que corresponde al idioma del texto del *Tweet* (`text`) detectado automáticamente por Twitter.

Existen otros atributos raíz que dan información adicional sobre el trino como `favorite_count` (el número de *"Me gusta"* que ha tenido el *tweet*), `retweet_count` (número de veces que el *tweet* ha sido reproducido), los cuales proporcionan información valiosa dependiendo del contexto y del objetivo del proyecto [8].

En este trabajo, un propósito fundamental es poder caracterizar un *tweet* a través de estos atributos y a partir de sus valores poder escoger los más adecuados para su procesamiento mediante el algoritmo de minería de datos. Para ello, se requiere una *aplicación* que interactúe con el API de Twitter para recolectar datos. Dicha aplicación está escrita en *Python*.

2.4 El lenguaje de programación Python

Python es un lenguaje de programación creado por el científico holandés Guido van Rossum en 1990. Sus principales características son las siguientes: es un lenguaje de *alto nivel* (similar al idioma inglés); *interpretado* (los programas se ejecutan directamente a través de un intérprete sin necesidad de compilación); *multiparadigma* (acepta diversas técnicas de programación); *de tipado dinámico* (las variables no necesitan ser declaradas o definidas antes de su uso, sino que su valor se asigna en tiempo de ejecución); *multiplataforma* (puede ser interpretado en gran variedad de sistemas operativos como Windows, Mac OS, Linux, Unix, Solaris, etc.) [9].

Las características anteriores, unidas a una filosofía de código legible, han hecho de *Python* uno de los lenguajes más utilizados en el ámbito académico y empresarial. Se considera un lenguaje maduro, con una base importante de programadores, documentación y proyectos en desarrollo [10].

Este lenguaje consta de *librerías* o grupos de subrutinas o programas que permiten realizar tareas complejas escribiendo pocas líneas de código. También puede contener tipos de datos específicos y procedimientos para crear nuevos tipos de datos.

En *Python* se escriben los programas o *aplicaciones* que recolectan datos desde la API de Twitter. Sin embargo, es necesario garantizar el *almacenamiento* o *persistencia* para realizar análisis posteriores, para lo cual se requiere un *sistema de administración de bases de datos* (SABD) como *PostgresSQL*.

2.5 El motor de base de datos PostgreSQL/PostGIS

PostgreSQL es un *sistema de administración de bases de datos*, caracterizado por ser *de propósito general* y *relacional*. Es de código abierto y gratuito. Emplea una variante del *lenguaje de consulta SQL*.

Entre sus ventajas se encuentran el ser 100% ACID, soporta distintos tipos de datos y alta concurrencia (esta última característica lo hace ideal para propósitos multiusuario). También soporta grandes volúmenes de datos [11].

Algunas desventajas menores tienen que ver con los errores en las transacciones, los cuales causan que se aborte toda la operación.

En *PostgreSQL* se crea la base de datos que almacenará los *tweets* recolectados mediante la aplicación de *Python* conectada a la API de Twitter.

Un complemento importante de *PostgreSQL* es *PostGIS*, el cual almacena todos los tipos de datos y funciones *espaciales*, que implican el almacenamiento de *geometrías* (puntos, líneas, polígonos, colecciones, etc.). A través de las *funciones espaciales*, estas geometrías pueden relacionarse entre sí, obteniendo nueva información que es útil para la toma de decisiones.

2.6 El *software* ArcMap® de ESRI

ArcMap es actualmente la herramienta de información geográfica más utilizada en el mundo occidental. Es propiedad de la empresa *ESRI* y se ha afianzado en los últimos años como el líder indiscutible del desarrollo de aplicaciones de manejo de información geográfica.

Este software viene provisto de numerosas *herramientas de análisis espacial*, de las cuales se emplearán algunas relacionadas con la *estadística*

espacial y los *métodos de agrupamiento*, fundamentales para la etapa de modelamiento propuesta en la metodología CRISP-DM®. Se destacan las siguientes:

2.6.1 *Densidad Kernel (Kernel Density)*

La herramienta *Densidad Kernel* calcula un modelo *raster* donde el valor de cada celda representa la densidad de puntos (número de entidades de punto por unidad de área), *en la vecindad de cada punto* [12].

El método opera determinando el número de entidades de punto que se encuentran en un *radio de búsqueda* (*Search Radius*) definido por la siguiente ecuación [12]:

$$SearchRadius = 0.9 * \min\left(SD, \sqrt{\frac{1}{\ln(2)}} * D_m\right) * n^{-0.2} \quad (1)$$

SD: distancia estándar ponderada (medida del grado de *concentración* o *dispersión* de entidades alrededor del centro medio geométrico [13].

D_m*:* mediana de la distancia desde el centro medio para todos los puntos

n: número de puntos

min hace referencia a que se emplea cualquiera de las dos opciones en el paréntesis que dé el menor valor.

2.6.2 *Análisis de Puntos Calientes Optimizado (Optimized Hotspot Analysis)*

Esta función de *ArcMap* identifica *clusters* o aglomeraciones de entidades de punto *estadísticamente significativas* de valores altos (*puntos calientes*) o valores bajos (*puntos fríos*).

El análisis se enfoca en *la presencia o ausencia del evento*, por lo cual resulta útil para determinar la probabilidad de aparición de una entidad de punto en un espacio dado [14].

La herramienta funciona definiendo una capa de incidentes o de eventos y unos *polígonos de delimitación donde es posible que ocurran incidentes*, empleando la *estadística Gi* de Getis-Ord*, la cual considera la búsqueda de cada entidad dentro del contexto de entidades vecinas. Un punto caliente estadísticamente significativo se caracteriza por ser una entidad de valor alto y por estar rodeada de valores igualmente altos. La misma consideración, con valores bajos, aplica para encontrar un punto frío. Los puntos calientes y fríos son, según Gi*, resultado de un fenómeno distinto a algo puramente aleatorio [15], [16].

La estadística Gi* es una *puntuación Z* (desviación estándar), donde valores altos y positivos indican un *punto caliente*, valores negativos (de valor absoluto alto) indican un *punto frío* y valores cercanos a cero indican una *falta de significancia estadística* (el fenómeno es aleatorio). Estos valores son asignados a los *polígonos de delimitación* [16].

3. Aplicación de la Metodología CRISP-DM

En el desarrollo de este trabajo, se empleará la metodología CRISP-DM® a datos recolectados desde Twitter. Para ello, se implementarán cada una de las etapas propuestas.

3.1 Comprensión del negocio

En esta fase de la metodología, se dará respuesta a cada una de las cuestiones planteadas a continuación.

3.1.1 *Objetivos del negocio.*

La georreferenciación de datos en Colombia provenientes de nuevas fuentes de información como las redes sociales es un campo poco explorado y de un gran potencial para las entidades públicas y privadas. En este contexto, el *negocio* consiste en la adquisición de *conocimiento geográfico útil* (aquel que permita la toma de decisiones o el planteamiento de nuevas investigaciones) a partir de dichas fuentes, en especial una que dé un acceso relativamente libre a los datos (como *Twitter*) a través de una aplicación escrita en un lenguaje de programación bien documentado y de código libre (*Python*), que permita almacenar la información en una base de datos robusta y de uso gratuito (*Postgres*) y que, luego de un proceso de limpieza y depuración de los datos, haga posible el uso de herramientas de análisis en software especializado como *ArcMap* para encontrar patrones de distribución espacial, que revelarán conocimiento no evidente a partir de los datos crudos.

3.1.2 *Valoración de la situación.*

La forma y el contenido de la información georreferenciada desde Colombia (y específicamente desde Bogotá) puede extraerse en *tiempo real* empleando la API de Twitter, un proceso relativamente rápido

teniendo en cuenta los actuales recursos informáticos disponibles (equipos y conexiones a Internet de alta velocidad, librerías o conjuntos de funciones especializadas en los lenguajes de programación, grandes capacidades de las bases de datos, etc.).

Por otra parte, los algoritmos de minería de datos (análisis espacial) se llevan a cabo en software especializado como *ArcMap*. Sin embargo, existen algoritmos similares en otras herramientas de información geográfica como *QGIS* o *gvSIG* mediante licencias otorgadas por la Universidad Militar o de acceso gratuito. En conclusión, la minería de datos sobre los datos georreferenciados provenientes de *Twitter* es viable.

3.1.3 *Objetivos de la minería de datos*.

Los objetivos del negocio en función de la minería de datos que se realizará son: 1. Visualizar los puntos correspondientes a *tweets* georreferenciados en una herramienta de información geográfica. 2. Determinar el porcentaje de tweets georreferenciados por localidad. 3. Determinar la *densidad de tweets* georreferenciados. 4. Determinar las localidades, sectores catastrales (barrios) y manzanas desde las que más *probabilidad* existe de encontrar un *tweet*.

3.1.4 *Plan de Proyecto.*

En el proyecto se contemplan las siguientes etapas generales: 1. *Recolección de datos desde Twitter empleando un programa o aplicación escrita en Python.* La recolección incluye una depuración inicial y el almacenamiento en una base de datos de Postgres. 2. *Exploración y verificación de calidad de los datos.* 3. *Preparación de los datos para el análisis*, lo que implica principalmente eliminación de datos que se encuentren fuera del perímetro urbano de Bogotá. 4. *Modelado*. En esta etapa se utilizarán las herramientas de análisis espacial provistas por *ArcMap* para encontrar los patrones y tendencias *ocultas* en la información recolectada. 5. *Evaluación de los resultados*. Se evaluarán los resultados de los análisis a la luz de la evidencia existente (datos originales). 6. *Implementación*. Para este trabajo, se plantearán algunas recomendaciones basadas en los resultados de los análisis. De la misma forma, se dará lugar a nuevas inquietudes que son a su vez la semilla de otros proyectos de minería de datos.

3.2 Comprensión de los datos

La fase de comprensión de los datos incluye la recolección, descripción, exploración y verificación de la calidad de los mismos.

3.2.1 *Recolección de los datos*

Se emplea la aplicación `twitterStreaming.py`, escrita en lenguaje *Python* (versión 2.7.15) y basada en las referencias [17], [18] y [19], cuyos elementos fundamentales son los siguientes:

Importación de librerías. En el encabezado del programa se emplean las librerías que se muestran en la figura No. 2.

```
import twitterCredentials

import psycopg2

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

import json
import csv
from datetime import datetime
from datetime import date
```

Figura 2. Librerías utilizadas en la aplicación de *Python* para la recolección de Tweets (Fuente: *elaboración propia*)

Es preciso destacar las siguientes librerías:

`tweepy`, librería con funciones que permiten comunicarse con la API de Twitter. Se emplean los módulos `StreamListener` y `Stream` para captar permanentemente datos desde el punto de conexión de Twitter.

`psycopg2`, permite conectarse a una base de datos alojada en *Postgres* y ejecutar sentencias SQL para ingresar información o realizar consultas.

`json`, librería que permite manipular datos en el formato nativo de Twitter.

Las librerías `csv` y `datetime` hacen parte del conjunto básico de funciones de *Python* y permiten escribir archivos de texto plano csv y manejar fechas, respectivamente.

Importación de credenciales. *Twitter* no permite un acceso completamente libre a los datos. Es necesario crear una cuenta de desarrollador, en cuya solicitud se requiere explicar los alcances y el tipo de datos empleados por las *aplicaciones* que el programador vaya a crear. Una vez superado este filtro, se tiene acceso a la información histórica y en tiempo real de la red social a través de unas *credenciales* que otorga el sistema y que son solicitadas en cada conexión.

Para efectos de seguridad, las credenciales originales se almacenan en un archivo de Python denominado `twitterCredentials.py` que acompaña al código fuente y se invocan desde éste con el siguiente bloque de instrucciones:

```
consumer_key = twitterCredentials.login['consumer_key']
consumer_secret = twitterCredentials.login['consumer_secret']
access_token = twitterCredentials.login['access_token']
access_token_secret twitterCredentials.login['access_token_secret']
```

Figura 3. Importación de credenciales de acceso a datos de Twitter desde `twitterCredencials.py` (Fuente: *elaboración propia*)

Persistencia de la información recolectada. El siguiente bloque en el programa consiste en definir cómo se almacenan los datos. Para efectos de este trabajo, se guardarán en tablas, para lo cual es necesario conectar a la aplicación con una base de datos previamente creada en *PostgreSQL*:

```
dbName = 'tweets'
u = 'postgres'
pw = 'SIG'
h = 'localhost'

try:
    conn = psycopg2.connect(database = dbName,
                            user = u,
                            password = pw,
                            host = h)
    cur = conn.cursor()
except Exception as e:
    print("Error en la conexión a la base de datos",
str(e))
```

Figura 4. Instrucciones de conexión a la base de datos de Postgres mediante funciones de la librería `psycopg2` (Fuente: *elaboración propia*)

En el sistema se crea la tabla `geotweets_YYmmdd` (que almacena los *tweets* georreferenciados por usuarios y YYmmdd indican el año, mes y día de creación de la entidad). En la figura 5 se muestran las instrucciones para la generación de esta tabla en la base de datos, con los atributos `count` (contador de tipo entero, llave primaria), `date` (fecha de creación del *tweet*), `tweetText` (texto del *tweet*), `source` (fuente del *tweet*), `language` (idioma del texto detectado automáticamente por Twitter), `place` (lugar asociado al *Tweet*), `lon` (posición geográfica en x -longitud), `lat` (posición geográfica en y - latitud), coordenadas en sistema EPSG 4326 (WGS84).

Adicionalmente, se crea la columna `geom` para almacenar la *geometría* de los puntos a través de la consulta `SELECT addGeometryColumn` y de esta forma poder visualizarlos en un sistema de información geográfica.

```
tableName1 = "geotweets_" +
datetime.today().strftime('%Y%m%d')

createTable1 = """CREATE TABLE {table}
                  (count INTEGER,
                   date TIMESTAMP WITH TIME ZONE,
                   tweet text CHARACTER VARYING(1000),
                   source CHARACTER VARYING(250),
                   language CHARACTER VARYING(20),
                   place CHARACTER VARYING (100),
                   lon DOUBLE PRECISION,
                   lat DOUBLE PRECISION,
                   PRIMARY KEY (count));""".format(table
= tableName1)


addGeometryColumn1 = ("SELECT AddGeometryColumn('"
                      + tableName1
                      + "','geom',4326,'POINT',2);")

try:
    cur.execute(createTable1)
    cur.execute(addGeometryColumn1)
    conn.commit()

except Exception as e:
    print("Hubo un error creando las tablas en la base
de datos", str(e))
```

Figura 5. Instrucciones de creación de la tabla `geotweets_YYmmdd` (Fuente: *elaboración propia*)

Recolección de los *Tweets*. La recolección de los objetos *tweets* (o *tweet objects*) se realiza a través de una *clase*, entidad que reúne un conjunto de *métodos* o subrutinas para realizar una determinada acción. La clase se denomina `TweetListener` y su función es gestionar el manejo de cada objeto que llega desde el API de Twitter *en modo streaming* o en *tiempo real*. Esta gestión incluye los siguientes pasos:

Almacenar cada objeto *tweet* en una variable tipo `json` llamada `json_tweet_data`.

Extraer de la variable anterior la información que es relevante de cada *tweet* asignándola a nuevas variables: texto del tweet (`tweetText`), coordenadas del tweet (`coordinates`), fuente del tweet (`tweetSource`), fecha de creación del tweet (`tweetDate`), idioma del tweet (`tweetLang`) y lugar asociado al tweet (`tweetPlace`).

Realizar un primer filtro de los tweets que serán almacenados en la base de datos, utilizando el condicional:

```
if coords is not None and tweetCountry == 'Colombia':

   tweetLon = coords["coordinates"][0]
   tweetLat = coords["coordinates"][1]

   geomPointQuery = ("ST_GeomFromText('POINT("
                     + str(tweetLon) + " "
                     + str(tweetLat) + ")',4326)")
```

Figura 6. Bloque condicional de asignación de valores de longitud y latitud (Fuente: *elaboración propia*)

Este bloque de instrucciones establece que, si el atributo `coordinates` del objeto *tweet* no es nulo y el lugar asociado corresponde a Colombia, se asignan los valores de longitud y latitud a sus respectivas variables y posteriormente se establece la consulta SQL que permite almacenar estos valores como *geometría* en la base de datos.

Finalmente, se ejecuta la sentencia SQL de inserción de los valores en las tablas, como se muestra en la figura 7.

```
sqlInsertIntoTable = ("INSERT INTO " + tableName1
                      + " VALUES (" + str(geoTweetsCounter)
                      + ",%s,'"
                      + tweetText+ "','"
                      + tweetSource + "','"
                      + tweetLang+ "','"
                      + tweetPlace + "',"
                      + str(tweetLon) + ","
                      + str(tweetLat)+ ","
                      + geomPointQuery + ");")
try:
  cur.execute(sqlInsertIntoTable,(str(tweetDate),))
  conn.commit()

except Exception as e:
  print("Hubo un error escribiendo en la tabla de tweets
georreferenciados", str(e))
```

Figura 7. Bloque de ejecución de sentencia SQL de inserción de datos en una tabla (Fuente: *elaboración propia*)

Por cada dato incluido en la tabla se incrementa el valor de un contador, que también es el ID de cada registro.

En el bloque principal del programa, la clase `TweetListener` se instancia en el objeto `listener` y posteriormente se accede a la API (y a los datos) de Twitter mediante las funciones que provee la librería `tweepy`, como se observa en la figura 8:

```
if __name__ == "__main__":

    listener = TweetListener()

    auth = OAuthHandler(consumer_key,consumer_secret)
    auth.set_access_token(access_token,access_token_secret)

    stream = Stream(auth,listener)

    bBox = [-74.286431,4.462925,-73.974975,4.829433] #Bogotá

    stream.filter(locations=bBox)
```

Figura 8. Bloque principal de la aplicación de recolección de datos de Twitter (Fuente: *elaboración propia*)

En este bloque de instrucciones, se destaca el *criterio de consulta* de los tweets: un *rectángulo georreferenciado* (*bounding box*) con dos puntos: (-74.286431; 4.462925) y (-73.974975; 4.829433), correspondientes a las esquinas inferior izquierda y superior derecha del rectángulo, respectivamente (Figura 9). Dentro de este rectángulo, que cubre la totalidad del perímetro urbano de la ciudad de Bogotá, se realiza la búsqueda de los *tweets* generados en tiempo real.

Figura 9. Rectángulo (*bounding box*) dentro del que se realiza la búsqueda de *tweets* georreferenciados (Fuente: *elaboración propia*)

El programa completo se puede consultar en el Anexo No. 1.

3.2.2 *Descripción de los datos*

El programa anteriormente descrito se ejecutó entre los días 22 de noviembre y 12 de diciembre de 2018, obteniendo varios conjuntos de datos correspondientes a fechas distintas.

Para este trabajo se escogió la muestra con mayor cantidad de datos recolectados, tomada desde las 12:10 horas del 24 de noviembre de 2018 hasta las 0:06 horas del 27 de noviembre de 2018, cuando se interrumpió la recolección por fallas en el servicio de Internet. En total se recolectaron 1841 *tweets* georreferenciados, almacenados en la tabla `geotweets_20181124`.

La tabla 1 ilustra un registro típico de la entidad `geotweets_20181124`.

TABLA No. 1 Registro típico almacenado en la tabla `geotweets_20181124`	
`count`	431
`date`	2018-11-24 21:25:13-05
`tweetText`	Cuando caminas y por un segundo miras a tu alrededor en Suba, Cundinamarca, Colombia https://t.co/vVbWdFoxmb
`source`	Instagram
`language`	Es
`place`	Bogotá, D.C., Colombia
`lon`	-74.0833
`lat`	4.75
`geom`	0101000020E61000006FF085C9548552C000000000000001340

(Fuente: *elaboración propia*)

Los atributos `geom`, `lon` (longitud o coordenada x) y `lat` (latitud o coordenada y) permiten visualizar los registros de la tabla como puntos en un *software* de manejo de información geográfica como *ArcMap* al conectarse a la base de datos `tweets`. En la figura 10 se muestran los puntos recolectados dentro del perímetro de la ciudad de Bogotá.

A partir de la tabla `geotweets_20181124` y de los registros visualizados en un sistema de información geográfica, en general se observa lo siguiente:

- Algunos registros no se encuentran dentro del rectángulo de búsqueda definido en el programa `twitterStreaming.py`.
- Los *tweets* georreferenciados tienden a aglomerarse hacia el centro y oriente de la ciudad de Bogotá.
- La mayoría de *tweets* provienen de fuentes como *Instagram* (aplicación para publicación de fotografías o videos) o *Foursquare* (aplicación que permite georreferenciar lugares –sobre todo negocios-, otorgando puntos por el descubrimiento de nuevos lugares).
- Existen algunos *tweets* localizados muy lejos de Bogotá.

Las observaciones anteriores pueden cuantificarse realizando una *exploración de los datos*, siguiente paso planteado en la metodología CRISP-DM.

3.2.3 *Exploración y Verificación de los datos*

La exploración de los datos recolectados contempla los siguientes aspectos:

Fuente de los *tweets*. La tabla No. 2 muestra la cantidad de *tweets* por fuente y su respectivo porcentaje.

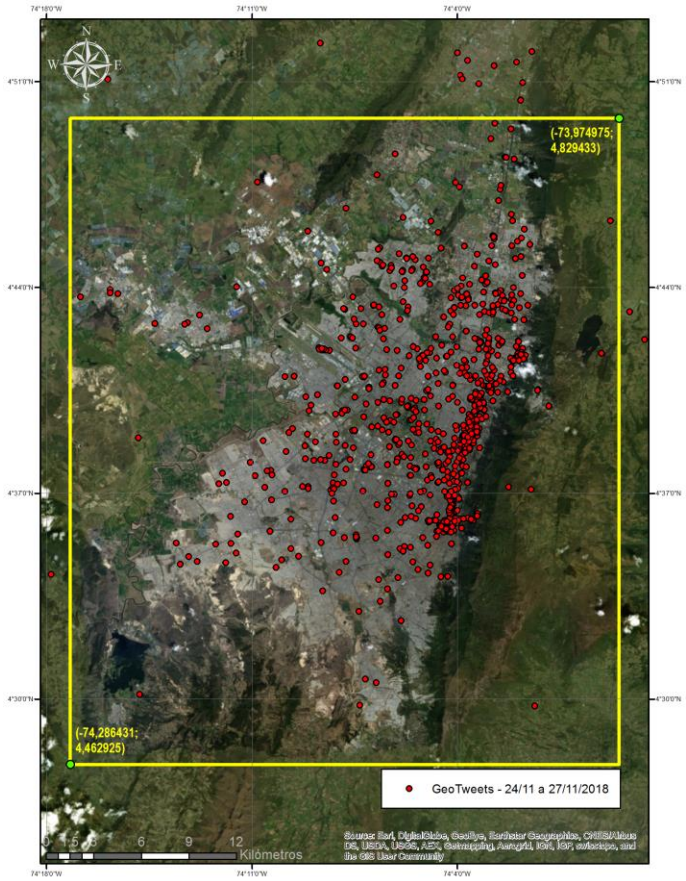

Figura 10. T*weets* georreferenciados captados del 24 al 27 de noviembre de 2018 empleando `twitterStreaming.py`
(Fuente: *elaboración propia*)

TABLA No. 2 Porcentaje de *tweets* almacenados en la tabla `geotweets_20181124` según fuente		
FUENTE	**NÚMERO DE TWEETS**	**PORCENTAJE**
Instagram	1400	76,04%
Foursquare	250	13,58%
Twitter for Android	97	5,27%
Twitter for iPhone	5	0,27%
Otras	87	4,73%

(Fuente: *elaboración propia*)

Es evidente que la gran mayoría de *tweets* localizados corresponden a publicaciones en la red social *Instagram* que a su vez se comparten a través de Twitter. El siguiente lugar lo ocupa el servicio de geolocalización *Foursquare*. Un 5,5% corresponde a *tweets* geolocalizados directamente desde aplicaciones de

dispositivos móviles, y el porcentaje restante se reparte entre varios servicios (*Endomondo, Career Arc 2.0, Tweetlogix, Boteando Trendy*, etc.).

Porcentaje de tweets localizados dentro del perímetro urbano de Bogotá. Mediante la herramienta *Select by Location* de ArcMap, es posible determinar la cantidad y el porcentaje de *tweets* geolocalizados dentro del perímero urbano de la ciudad de Bogotá, dado que solo interesan estos datos para el propósito de este estudio.

El resultado indica que para la muestra en estudio se encuentran 1669 registros localizados dentro del perímetro de la ciudad, lo que equivale a un 90,66% del total de *tweets* recolectados.

Porcentaje de tweets por localidad. Es de interés conocer cuántos *tweets* se generan teniendo en cuenta la división político-administrativa de la ciudad de Bogotá: las *localidades* que a su vez agrupan los *sectores catastrales* o barrios. Actualmente existen veinte (20) localidades, cuya delimitación está disponible en el *Mapa de Referencia del IDECA* (Infraestructura Integrada de Datos Espaciales para el Distrito Capital) [20].

Empleando la herramienta *Spatial Join* de ArcMap, se determinan cuántos puntos se encuentran dentro de un polígono (en este caso, la localidad).

De la tabla de porcentajes de *tweets* por localidad (Tabla 3) y del mapa (Figura 11), puede observarse lo siguiente:

- Los *tweets* georreferenciados se originan en su mayor parte desde las localidades de Chapinero y Suba.
- Los menores porcentajes de *tweets* geolocalizados corresponden a las divisiones administrativas del sur de la ciudad (en su orden: Tunjuelito, Ciudad Bolívar, Rafael Uribe Uribe, Usme y Sumapaz).
- La distribución espacial de los *tweets* se concentra hacia el norte y oriente de la ciudad, observándose aglomeraciones de puntos en las localidades de La Candelaria, Chapinero, Teusaquillo y Usaquén. En el resto de localidades los puntos se observan más dispersos.

TABLA No. 3
Número de *tweets* y porcentaje por localidad de Bogotá D.C.

CÓDIGO	NOMBRE	No. DE TWEETS	PORCENTAJE
1	USAQUEN	175	9,506
2	CHAPINERO	305	16,567
3	SANTA FE	114	6,192
4	SAN CRISTOBAL	14	0,760
5	USME	3	0,163
6	TUNJUELITO	6	0,326
7	BOSA	12	0,652
8	KENNEDY	60	3,259
9	FONTIBON	103	5,595
10	ENGATIVA	120	6,518
11	SUBA	296	16,078
12	BARRIOS UNIDOS	67	3,639
13	TEUSAQUILLO	139	7,550
14	LOS MARTIRES	42	2,281
15	ANTONIO NARIÑO	17	0,923
16	PUENTE ARANDA	56	3,042
17	CANDELARIA	129	7,007
18	RAFAEL URIBE URIBE	4	0,217
19	CIUDAD BOLIVAR	7	0,380
20	SUMAPAZ	0	0,000

(Fuente: *elaboración propia*)

Los resultados de la anterior *exploración descriptiva* de los datos recolectados han permitido hallar información *no evidente* sobre los *tweets*. Sin embargo, surgen las siguientes preguntas que no pueden ser respondidas con un análisis meramente descriptivo:

- ¿Dónde se concentran más los *tweets* en Bogotá?
- ¿Dónde es más *probable* encontrar *tweets* geolocalizados en Bogotá?
- ¿Por qué los *tweets* en ciertos lugares están aglomerados y en otros no?

La respuesta a estas cuestiones se obtiene mediante *algoritmos de agrupamiento* con *significado estadístico*, tema que en la metodología CRISP-DM se aborda en el *modelado*.

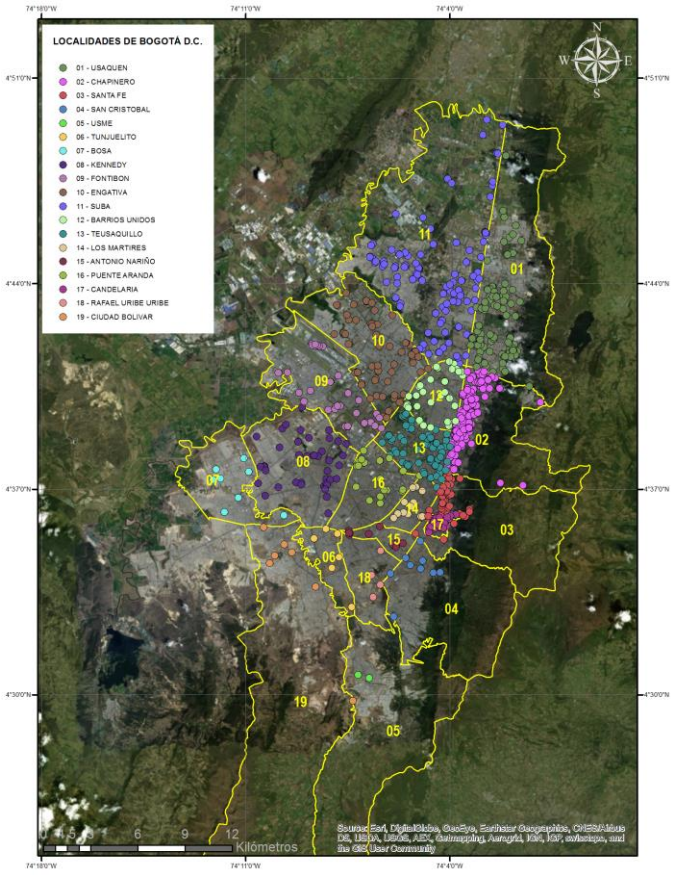

Figura 11. T*weets* georreferenciados diferenciados por localidad
(Fuente: *elaboración propia*)

3.3 Preparación de los datos

La *preparación de los datos* tuvo lugar durante las fases de recolección y exploración, al implementar los siguientes procedimientos:

3.3.1 *Recolección de tweets con atributo* `place` *en Colombia*

En el condicional mostrado en la figura 6 se restringió el país (atributo `place['country']`) del que proviene el *tweet* a Colombia, con el fin de evitar la recolección de datos provenientes de otros lugares (especialmente de Brasil) observada en muestras tomadas anteriormente.

3.3.2 *Descarte de tweets georreferenciados de alta repetición*

Algunos de los *tweets* geolocalizados se ubican siempre en el mismo punto. Esto es normal, teniendo en cuenta que *Twitter* predetermina la ubicación de algunos lugares (como barrios, sitios de interés, ciudades, etc.).

Sin embargo, hay dos pares de coordenadas que se repiten constantemente en las pruebas preliminares, las que finalmente se decidió eliminar para no generar una sobrepoblación de estos puntos, con miras a utilizar los datos en el proceso de modelación. Los dos pares de coordenadas se muestran en la tabla No. 4.

TABLA No. 4
Coordenadas de *tweets* georreferenciados de alta repetición

TEXTO O ASUNTO DEL TWEET	LONGITUD (°)	LATITUD (°)	CONTEO
Ubicación predeterminada de Bogotá, Colombia en Twitter e Instagram	-74,0794	4,5997	1395
Punto correspondiente a *tweets* geolocalizados por el servicio *Trendsmap Alerting*	-74,082	4,60987	148

(Fuente: *elaboración propia*)

Los *tweets* que se comparten a través de una aplicación como *Instagram* y en donde solo se especifica que la ubicación es "Bogotá, Colombia", se georreferencian en las coordenadas (-74,0794; 4,5997). Por otra parte, el servicio *Trendsmap Alerting* (página web que permite conocer los *tweets* más populares, así como las tendencias o *trends*) genera mensajes con el texto "`es ahora una tendencia en Bogotá/Colombia`" y siempre emplea las coordenadas (-74,082; 4,60987). Estos puntos no corresponden a ningún sitio de interés, solo están ubicados dentro del perímetro urbano de Bogotá.

La muestra de datos tomada desde el 24 hasta el 27 de noviembre de 2018 registró 1543 *tweets* de alta repetición, los cuales sin duda alguna afectarían cualquier clase de cálculo estadístico, sesgando considerablemente la tendencia espacial. Por esta razón se eliminan de la tabla de *tweets* georreferenciados, en aras de tener mayor variabilidad en los registros.

Existen otros *tweets* con ubicaciones geográficas duplicadas. Sin embargo, a diferencia de los registros de alta repetición, corresponden a realidades geográficas bien definidas (barrios, centros comerciales, aeropuertos, sitios turísticos, etc.), de manera que, al repetirse, se comprueba que hay más usuarios en esa ubicación, no es simplemente una coordenada generada de manera automática.

El descarte de los *tweets* de alta repetición se realiza desde la aplicación de recolección de datos, al establecer dos procedimientos de búsqueda: uno encuentra los

tweets geolocalizados cuyo texto contenga la frase "`es ahora una tendencia`" y el otro encuentra los registros con coordenadas (-74,0794; 4,5997). Los *tweets* no se incluyen en la tabla final de registros georreferenciados empleando un condicional.

3.3.3 *Descarte de tweets georreferenciados fuera del perímetro urbano de Bogotá*

Para el conteo de *tweets* por localidades se descartaron los registros georreferenciados fuera del perímetro urbano de Bogotá D.C., con el fin de establecer un porcentaje exacto de *tweets* en cada localidad, haciendo que la totalidad (100%) de registros estuvieran en Bogotá; y, por otra parte, se cumple un requisito fundamental para la aplicación del modelo de puntos calientes: los *eventos* o puntos deben estar contenidos en los polígonos de agregación (localidades o barrios).

3.4 Modelado

El *modelado* consiste en la implementación de los *algoritmos de minería de datos* necesarios para responder las preguntas surgidas en las fases de conocimiento del negocio y de exploración de los datos.

La primera pregunta a responder es: *¿en qué localidades se registra la mayor concentración de tweets por unidad de área?* Esta cuestión surge al observar que los *tweets* se aglomeran en ciertas zonas de la ciudad (como en las localidades de La Candelaria y Chapinero) y en otras son numerosos, pero relativamente separados (como en Suba), por lo que un conteo de *tweets* por localidad no es suficiente para caracterizarlos espacialmente. Es necesaria una medida de *densidad por unidad de área* no solo para determinar si en una zona de Bogotá en particular se generan muchos *tweets* sino también si se encuentran dispersos o aglomerados. Esto constituye una caracterización *descriptiva* del fenómeno de geolocalización de *tweets* (correspondiente al instante o ventana de tiempo) y permite diferenciar claramente las zonas donde se produce tal fenómeno de aquellas en donde no se produce o está muy disperso.

Lo anterior conduce a la segunda pregunta a responder mediante el modelado: *¿en qué sectores catastrales (barrios) de Bogotá es más probable encontrar un tweet georreferenciado?* Para tal efecto se emplea la herramienta conocida como *Análisis Optimizado de Puntos Calientes*, la cual permite hallar aglomeraciones o ausencias de eventos estadísticamente significativas y estimar una *probabilidad* de encontrar un evento (*tweet*) en un polígono de agregación asociado (los barrios o sectores catastrales de Bogotá). Esta herramienta, a diferencia de la densidad por unidad de área, es de tipo *predictivo* y permite saber en qué zonas de Bogotá podría encontrarse un *tweet en cualquier instante de tiempo*.

3.4.1 Densidad *Kernel*

Para obtener el mapa de densidad de *tweets* por unidad de área se empleó la herramienta *Kernel Density* de ArcMap, en la que se especificaron los siguientes parámetros de entrada:

- *Tamaño de celda de salida:* 1 ha (10000 m^2, el tamaño promedio de una manzana catastral de Bogotá).
- *Radio de búsqueda:* 1378,99 metros (calculado con la ecuación (1) de manera automática por la herramienta).

De esta forma, se pretende medir la *densidad de tweets por hectárea*.

En la figura 11 se muestra la distribución de densidad obtenida, superpuesta a la capa de localidades de Bogotá. El valor máximo de densidad obtenido es de 0,2612 *tweets*/ha, correspondientes a la localidad de La Candelaria.

La densidad por hectárea permite detectar los siguientes hechos, no evidentes por sí mismos en la *información cruda* proveniente de Twitter:

- Los *tweets* geolocalizados se concentran en las localidades de La Candelaria (17) y Chapinero (02), en una franja que va desde la calle 6 (límite sur de La Candelaria) hasta la calle 104 (localidad de Usaquén), paralela a la Avenida Caracas y a la Autopista Norte.
- Dentro de la franja de mayor densidad, se encuentran tres regiones donde la concentración de *tweets* es mayor. La primera se ubica sobre La Candelaria, cuyo centro geográfico corresponde aproximadamente a la Plaza de Bolívar, centro histórico y turístico de Bogotá.
- La segunda región se encuentra en la localidad de Chapinero, más concretamente entre las calles 63 y 68 y Carreras 7 y 13 (Av. Caracas), con centro en el tradicional Parque de Lourdes.
- La tercera región de alta densidad corresponde a la llamada *Zona Rosa* de Bogotá, entre calles 78 y 88 y

carreras 11 y 15, con centro en la carrera 13 con calle 82, parte de la famosa *zona T* de restaurantes en Bogotá.

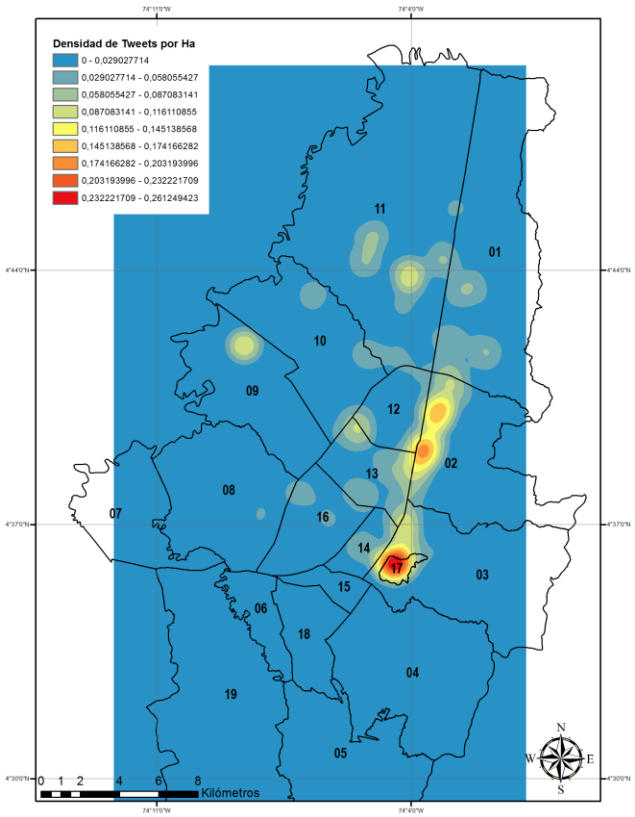

Figura 12. Densidad de *tweets* georreferenciados por hectárea, obtenida mediante la herramienta *Kernel Density* de ArcMap
(Fuente: *elaboración propia*)

Existen otras regiones de alta densidad de *tweets*, localizadas en el Parque Metropolitano Simón Bolívar (localidad 13 de Teusaquillo), el aeropuerto Luis Carlos Galán (antiguo Eldorado, localidad 09 de Fontibón) y el sector de Colina Campestre en la localidad 11 de Suba.

Lo anterior indica que, más allá de que se generen *tweets* en una determinada localidad, existen zonas específicas de la ciudad donde estos eventos se dan con mayor frecuencia, y dichas zonas coinciden con los *sitios turísticos, de diversión y de negocios* de la ciudad de Bogotá. Sobre este punto se tratará más detalladamente en el *análisis y evaluación* de los resultados obtenidos.

3.4.2 Análisis Optimizado de Puntos Calientes

Este algoritmo de *data mining* busca encontrar una *significancia estadística* en los eventos asociados a unos *polígonos de agregación* en los cuales se determina la *probabilidad* de encontrar el evento. En este caso, se considera únicamente la *ubicación espacial* como criterio de evaluación.

El resultado son los polígonos de agregación con nuevos valores: *Z* (desviación estándar), *p*(probabilidad) y un valor *Gi Bin* (nivel de confianza), que mide la significancia estadística.

Los parámetros de entrada de la herramienta *Optimized Hot Spot Analysis* de ArcMap son los siguientes:

- **Capa de entrada:** *tweets* georreferenciados dentro del perímetro urbano de Bogotá.

- **Método de agregación de datos de incidentes o eventos:** *contar los incidentes dentro de polígonos de agregación* (`COUNT _INCIDENTS _ WITHIN _ AGGREGATION _ POLYGONS`).

- **Polígonos de agregación de incidentes.** Se empleará la capa de *sectores catastrales* (o barrios), dado que el modelo solo permite el uso de capas con más de 30 eventos (puntos) y más de 30 polígonos de agregación. La capa de sectores catastrales, al igual que la de localidades, proviene del *Mapa de Referencia del IDECA*.

Luego de ejecutar el algoritmo, se obtiene el *mapa de puntos calientes y puntos fríos* (Figura 13), en donde los polígonos (zonas catastrales o barrios) se clasifican según el valor *Gi Bin* obtenido, el cual presenta el siguiente rango de valores:

TABLA No. 5
VALORES Gi Bin OBTENIDOS DEL ANÁLISIS OPTIMIZADO DE PUNTOS CALIENTES (HOT SPOTS)

VALOR	INTERPRETACIÓN
+3	Punto Caliente, estadísticamente significativo con 99% de confianza
+2	Punto Caliente, estadísticamente significativo con 95% de confianza
+1	Punto Caliente, estadísticamente significativo con 90% de confianza
0	No es estadísticamente significativo
-1	Punto Frío, estadísticamente significativo con 90% de confianza
-2	Punto Frío, estadísticamente significativo con 95% de confianza
-3	Punto Frío, estadísticamente significativo con 99% de confianza

(Fuente: *elaboración propia*)

En el mapa claramente se distinguen tres zonas de la ciudad de Bogotá: una con *alta probabilidad* (valor Gi Bin de +3) de localizar un *tweet* georreferenciado

(situada en las localidades de Usaquén (01), Chapinero (02), Santa Fe (03), Suba (11), Barrios Unidos (12), Teusaquillo (13), Los Mártires (14) y La Candelaria (17); y otra zona, con valores de *Gi Bin* de -1 ó -2 y por tanto con *baja probabilidad* (con niveles de confianza del 90% y del 95%) de hallar un tweet georreferenciado (localidades 04 de San Cristóbal, 05 de Usme, 06 de Tunjuelito, 07 de Bosa, 08 de Kennedy, 18 de Rafael Uribe Uribe y 19 de Ciudad Bolívar). La tercera zona, correspondiente al resto de localidades de Bogotá (incluyendo a Sumapaz) no presentan significancia estadística (valor de *Gi Bin* igual a cero) en cuanto a la presencia o ausencia de *tweets* geolocalizados, considerándose un fenómeno puramente aleatorio.

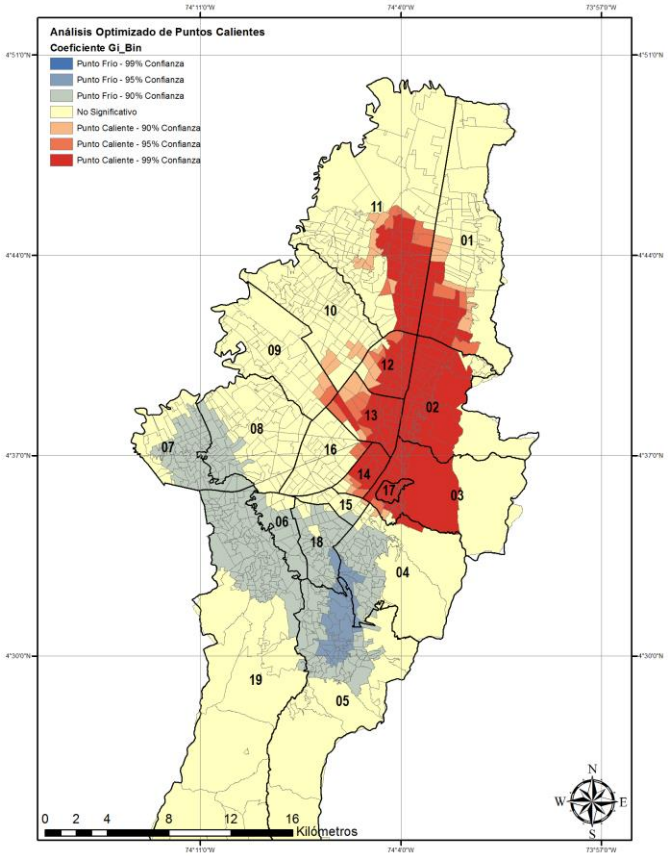

Figura 13. Sectores catastrales clasificados según el índice *Gi Bin* (significancia estadística de hallazgo de *tweets*), obtenida mediante la herramienta *Optimized Hot Spot Analysis* de ArcMap (Fuente: *elaboración propia*)

Dentro de las localidades, especialmente las de mayor área, también se observan importantes variaciones en la significancia estadística: en Suba y Usaquén, así como en Usme y Ciudad Bolívar, los sectores catastrales situados más cerca de la periferia de la ciudad presentan valores de *Gi Bin* iguales a cero. En contraste, localidades como La Candelaria y Teusaquillo están contenidas en la zona de alta significancia estadística, mientras que en Santa Fe y Chapinero se observan valores de *Gi Bin* de +3 a excepción de las zonas de protección ambiental de los Cerros Orientales.

Es notable que solo en una localidad (San Cristóbal) se encuentran tanto puntos calientes como puntos fríos y zonas de "cero significancia estadística". El resto de localidades presentan tendencia a albergar puntos calientes, puntos fríos, o a no ser significativas.

3.5 Evaluación

Los algoritmos empleados en el proceso de *modelado* han permitido encontrar una *respuesta* a las preguntas planteadas desde la fase de conocimiento del negocio, determinando *dónde* y *cómo* se distribuyen los *tweets* geolocalizados en Bogotá. A continuación, se expone una evaluación de los resultados generales, así como una revisión de los aspectos por mejorar.

3.5.1 *Evaluación de los resultados*

La fortaleza de los modelos empleados (densidad kernel y análisis optimizado de puntos calientes) estriba en su capacidad para obtener resultados *más allá* de lo puramente visual, estableciendo patrones que no son evidentes al ver los datos crudos o sin procesar.

Los hallazgos de la fase de modelamiento se sintetizan en los siguientes puntos:

Los *tweets* georreferenciados se generan en las zonas turísticas y de negocios de Bogotá. Más allá de ser un fenómeno de naturaleza aleatoria, los trinos o *tweets* geolocalizados en la ciudad de Bogotá se originan en áreas con una oferta elevada de sitios turísticos (museos, lugares de interés histórico), o servicios (restaurantes, discotecas, sector financiero). Esto se explica en el hecho de que el *tweet* georreferenciado está asociado en más del 90% de los casos a imágenes (publicadas a través de *Instagram*), o a localización de lugares de interés (*Foursquare*). Las personas que publican estos trinos publican imágenes de los sitios que visitan o desean generar algún valor por anunciar que se encuentran en un lugar que fue de su agrado, para de esta forma atraer más visitantes.

Un análisis cuantitativo y más detallado de la *intención* del *tweet* y su relación con la ubicación geográfica requiere un ejercicio denominado *análisis de sentimientos* o *análisis de opinión* sobre el contenido del *tweet*, que requiere otros algoritmos y procedimientos que están fuera del alcance de este trabajo.

Los patrones de *tweets* georreferenciados permiten detectar que la oferta turística y de servicios en Bogotá está limitada a un sector en particular. A partir del origen del *tweet* geolocalizado, el mapa de la figura 13 muestra un evidente sesgo de la oferta turística y de servicios hacia el centro y nororiente de la capital colombiana. La probabilidad de encontrar un *tweet* geolocalizado no es estadísticamente significativa o es muy baja hacia el sur y occidente de la ciudad, sobre todo en las localidades de Usme, Tunjuelito, Rafael Uribe Uribe y Ciudad Bolívar, consideradas de las zonas más pobres de la capital.

3.5.2 *Revisión y aspectos por mejorar*

En el proceso también se evidenciaron aspectos por mejorar, los cuales permitirán encontrar nuevos resultados.

Se requiere una muestra de mayor tamaño en número de eventos y ventana de tiempo para seguir analizando el fenómeno. Después del proceso de preparación y limpieza de los datos, la muestra final se redujo a 1661 eventos distribuidos en una ventana de tiempo de 67 horas. Esta muestra, aunque significativa en principio, requiere los siguientes ajustes para futuros análisis:

- La ventana de tiempo debe abarcar días laborales y fines de semana.

- Deben diferenciarse los resultados correspondientes a los días laborales y a los fines de semana.

Es necesario considerar la información proveniente de otros atributos raíz del objeto *tweet.* Los *retweets* o los *likes* (“me gusta”) de un objeto *tweet* son fundamentales para determinar si existen tendencias con respecto a un lugar, sector o incluso *marca*. Son un insumo valioso para análisis de opinión o de mercado.

3.6 Implementación

A partir de los resultados del modelo, surgen recomendaciones de aplicación de los resultados, ejecución del modelo en otros conjuntos de datos y preguntas que podrían generar nuevas investigaciones. A continuación, se presentan dichas recomendaciones.

Los *tweets* georreferenciados son una herramienta potencial para promocionar sitios o sectores aún desconocidos para el público. Para cualquier persona que vea por primera vez los mapas de las figuras 12 y 14, será evidente que “algo” ocurre en el oriente de la ciudad de Bogotá y que “nada” ocurre en el resto. Sin embargo, bien se sabe que Bogotá cuenta con otras atracciones turísticas y ecológicas y ofertas de servicios fuera del eje La Candelaria – Chapinero – Usaquén, tales como los humedales del occidente de la ciudad, las plazas centrales de las localidades de Bosa, Fontibón, Engativá y Suba, los parques de La Florida y Mirador de los Nevados, los centros comerciales Imperial y Titán Plaza, entre muchos otros sitios interesantes. El *tweet* georreferenciado se presenta así como un medio de promoción de lugares poco conocidos, que al visualizarse en un mapa despiertan el interés de potenciales visitantes.

Algunas empresas ya han comprendido el valor agregado del *tweet* geolocalizado y periódicamente promocionan sus productos adicionando su ubicación geográfica, para atraer más clientes y ser tendencia.

Una manera de generar más *tweets* geolocalizados desde un cierto lugar que se desee promocionar es dar un estímulo (descuentos, regalos, etc.) a las personas que publiquen una foto del sitio o actualicen su estado en Twitter o en *Foursquare* con la ubicación del lugar. Esa misma empresa podría publicar un servicio geográfico en su página web, donde muestre los *tweets* geolocalizados de la misma manera a como se realizó en este ejercicio y así generar interés sobre el sitio.

¿Qué marcas se repiten más en los *tweets* georreferenciados? Esta pregunta puede responderse mediante un análisis del *texto* del *tweet*, que se logra mediante una *tokenización* o división del texto en las palabras que lo componen. De esta forma podría generarse una estadística descriptiva básica y un análisis espacial de las *marcas* que aparecen con mayor frecuencia en los *tweets*, empleando esta información en análisis de mercado, etc.

¿En qué manzanas y calzadas se generan más *tweets* georreferenciados? Es posible realizar un análisis de los *tweets* no solo a nivel de toda la ciudad de Bogotá sino a nivel de localidades o incluso de sectores catastrales, para observar las tendencias a nivel local (que pueden ser de interés para una Junta Administradora Local –JAL– o una alcaldía menor), utilizando los polígonos de manzanas y calzadas disponibles en el mapa de referencia del IDECA.

¿La tendencia de encontrar *tweets* georreferenciados en sitios turísticos se repite para otras grandes ciudades colombianas? Este trabajo podría implementarse –con las debidas mejoras – en otras ciudades como Medellín, Cali o Barranquilla, únicamente modificando el rectángulo o *bounding box* de búsqueda de *tweets*, con el fin de verificar si la tendencia de ubicar

tweets en sitios turísticos se replica a nivel nacional o solo se da en la capital de la República.

4. Conclusiones

A partir de una aplicación escrita en lenguaje *Python*, desde el API de Twitter se recolectaron los *tweets* georreferenciados dentro de un área correspondiente al perímetro de Bogotá, durante 67 horas, del 24 al 27 de noviembre de 2018. El total de *tweets* recolectado fue de 1841, de los cuales se utilizaron 1669 registros para el análisis espacial al encontrarse dentro del perímetro urbano de Bogotá. Sobre estos datos se aplicaron los pasos descritos en la metodología CRISP-DM de minería de datos.

Los *tweets* georreferenciados presentan un patrón de aglomeración hacia el oriente y norte de la capital de Colombia, destacándose la localidad de La Candelaria como el principal *cluster* con una densidad de 0,23 *tweets* por hectárea. Asimismo, es más probable encontrar un *tweet* georreferenciado en una franja que va desde la Calle 6 a la Calle 104, entre la Avenida Caracas – Autopista Norte y la Carrera Séptima.

Por otra parte, las localidades ubicadas al sur de la ciudad (San Cristóbal, Usme, Tunjuelito, Bosa, Rafael Uribe Uribe y Ciudad Bolívar) presentan una baja probabilidad de ocurrencia de *tweets* geolocalizados.

Los *tweets* georreferenciados se asocian principalmente con *fotografías* y *localización* en sitios turísticos, gastronómicos, de entretenimiento (discotecas, centros comerciales) y de negocios, dado que el 89,62% provienen de fuentes como *Instagram* y *Foursquare*. Se encuentran tres zonas de alta densidad de *tweets* y de alta probabilidad de ocurrencia: la localidad de La Candelaria, Chapinero central (con centro en el Parque de Lourdes) y la Zona Rosa (con centro en la Zona T); las 3 zonas descritas son turísticas o de entretenimiento y negocios. Lo anterior evidencia una dinámica sesgada principalmente hacia el norte y oriente de la capital de la República.

Una recolección sistemática de datos provenientes de redes sociales permite obtener información no evidente u oculta, que genera conocimiento acerca de las dinámicas de los espacios donde dichos datos se generan.

Referencias

[1] González, J. (Noviembre de 2018) *Minería de Datos*, Universidad Politécnica de Puebla. Obtenido de: https://ccc.inaoep.mx/~jagonzalez/AI/Sesion13_Data_Mining.pdf

[2] León, E. (Noviembre de 2018) *Diplomado en Minería de Datos* (Universidad Nacional de Colombia). Obtenido de:http://disi.unal.edu.co/profesores/eleonguz/cursos/md/presentaciones/

[3] Ministerio de las TIC (Noviembre de 2017) *Colombia es uno de los países con más usuarios de las redes sociales en la región.* Obtenido de: https://www.mintic.gov.co/portal/604/w3-article-2713.html

[4] Gallardo, J. (2009) *Metodología para la Definición de Requisitos en proyectos de Data Mining.* Obtenido de: http://oa.upm.es/1946/1/JOSE_ALBERTO_GALLARDO_ARANCIBIA.pdf

[5] Web Empresa (Marzo de 2018) *¿Qué es Twitter? ¿Cómo funciona? ¿Cómo puedo usarlo para mi organización?* Obtenido de: https://www.webempresa.com/blog/que-es-twitter-como-funciona.html

[6] Twitter (2018) *Información sobre las API de Twitter.* Obtenido de: https://help.twitter.com/es/rules-and-policies/twitter-api

[7] Twitter (2018) *Introduction to Tweet JSON.* Obtenido de: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json

[8] Twitter (2018) *Tweet Object.* Obtenido de: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

[9] Bahit, E. (2013) *Python para Principantes.* Obtenido de: https://librosweb.es/libro/python/

[10] Paradigma Digital (Noviembre de 2017) *¿Es Python el lenguaje del futuro?* Obtenido de: https://www.paradigmadigital.com/dev/es-python-el-lenguaje-del-futuro/

[11] PostgreSQL Tutorial (2018) *What is PostgreSQL?* Obtenido de: http://www.postgresqltutorial.com/what-is-postgresql/

[12] ESRI (2018) *¿Cómo funciona la Densidad Kernel?* Obtenido de: https://pro.arcgis.com/es/pro-app/tool-reference/spatial-analyst/how-kernel-density-works.htm

[13] ESRI (2018) *Distancia Estándar*. Obtenido de: http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-statistics-toolbox/standard-distance.htm

[14] ESRI (2018) *Análisis de Puntos Calientes Optimizado*. Obtenido de: http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-statistics-toolbox/optimized-hot-spot-analysis.htm

[15] ESRI (2018) *¿Cómo funciona el análisis de puntos calientes optimizado?* Obtenido de: http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-statistics-toolbox/how-optimized-hot-spot-analysis-works.htm

[16] ESRI (2018) *¿Cómo funciona Análisis de Puntos Calientes (Gi* de Getis-Ord?* Obtenido de: http://desktop.arcgis.com/es/arcmap/10.3/tools/spatial-statistics-toolbox/h-how-hot-spot-analysis-getis-ord-gi-spatial-stati.htm

[17] Stone, N. (2017), *Social Media Canvassing Using Twitter and WebGIS to Aid in Solving Crime*, University of Southern California. Obtenido de: https://spatial.usc.edu/wp-content/uploads/2017/10/Stone_Neil.pdf

[18] Wageningen University and Research (WUR) Geoscripting (2018) *Harvesting Tweets with Python.* Obtenido de: http://geoscripting-wur.github.io/PythonWeek/PythonTwitter.html#Python_notebooks_for_the_self_study

[19] Pardo, P. (2013) *Geolocalizando Tweets con Python.* Obtenido de: http://pardozaragoza.blogspot.com/2013/04/capturando-tweets-con-python-tutorial-1.html

[20] IDECA (Septiembre de 2018) *Mapa de Referencia de Bogotá D.C.* Obtenido de: https://www.ideca.gov.co/es/servicios/mapa-de-referencia/tabla-mapa-referencia

Agradecimientos

El autor agradece la valiosa colaboración de las profesoras Marcela Mejía y Martha Liliana Quevedo, del Semillero de *Big Data Analytics* (Analítica de Datos Masivos) de la Universidad Militar Nueva Granada.

Anexo No. 1

Aplicación `twitterStreaming.py` para recolectar *tweets* georreferenciados desde Twitter®

```
#-------------------------------------------------------------
#UNIVERSIDAD MILITAR NUEVA GRANADA
#ESPECIALIZACIÓN EN GEOMÁTICA
#PROYECTO FINAL DE GEOMÁTICA APLICADA
#-------------------------------------------------------------

#Gustavo Adolfo García Vélez - 3101426

#Aplicación para recolectar objetos tipo tweet desde el API de Twitter y
#almacenarlos en una base de datos de Postgres, dada una ubicación
#geográfica definida por un rectángulo o bounding box

#Aplicación basada en las siguientes referencias:
#https://spatial.usc.edu/wp-content/uploads/2017/10/Stone Neil.pdf
#http://geoscripting-wur.github.io/PythonWeek/PythonTwitter.html#Python notebooks for the self study
#http://pardozaragoza.blogspot.com/2013/04/capturando-tweets-con-python-tutorial-1.html


#-------------------------------------------------------------
# IMPORTACIÓN DE LIBRERÍAS
#-------------------------------------------------------------

import twitterCredentials                       #Credenciales de desarrollador de Twitter

import psycopg2                                 #Librería para conexión a Postgres


from tweepy.streaming import StreamListener     #Librería para conexión al API de Twitter
from tweepy import OAuthHandler
from tweepy import Stream

import json                                     #Librería para manejo de archivos json
import csv                                      #Librería para manejo de archivos csv
from datetime import datetime                   #Librería para manejo de fechas
from datetime import date

#-------------------------------------------------------------
# IMPORTACIÓN DE CREDENCIALES DE TWITTER
#-------------------------------------------------------------

consumer_key = twitterCredentials.login['consumer_key']
consumer_secret = twitterCredentials.login['consumer_secret']
access_token = twitterCredentials.login['access_token']
access_token_secret = twitterCredentials.login['access_token_secret']


#-------------------------------------------------------------
#CONEXIÓN A LA BASE DE DATOS DE POSTGRES
#-------------------------------------------------------------

dbName = 'tweets'
u = 'postgres'
pw = 'SIG'
h = 'localhost'

try:
    conn = psycopg2.connect(database = dbName,
                            user = u,
                            password = pw,
                            host = h)
    cur = conn.cursor()
except Exception as e:
    print("Error en la conexión a la base de datos", str(e))

#-------------------------------------------------------------
#CREACIÓN DE LA TABLA DE TWEETS EN LA BASE DE DATOS tweets
#-------------------------------------------------------------
```

```
#Nombre de la tabla
tableName1 = "geotweets_" + datetime.today().strftime('%Y%m%d')

#Instrucción de creación de la tabla
createTable1 = """CREATE TABLE {table}
                (count INTEGER,
                 date TIMESTAMP WITH TIME ZONE,
                 tweet_text CHARACTER VARYING(1000),
                 source CHARACTER VARYING(250),
                 language CHARACTER VARYING(20),
                 place CHARACTER VARYING (100),
                 lon DOUBLE PRECISION,
                 lat DOUBLE PRECISION,
                 PRIMARY KEY (count));""".format(table = tableName1)


#Adición de la columna de geometría tipo punto
addGeometryColumn1 = ("SELECT AddGeometryColumn('"
                      + tableName1 + "','geom',4326,'POINT',2);")


try:
    cur.execute(createTable1)
    cur.execute(addGeometryColumn1)
    conn.commit()

except Exception as e:
    print("Hubo un error creando las tablas en la base de datos", str(e))


#------------------------------------------------------------
#DEFINICIÓN DE ARCHIVO CSV (PERSISTENCIA 2)
#------------------------------------------------------------

path1 = 'geoTweets_' + datetime.today().strftime('%Y%m%d') + '.csv'

#Create csv file
csvFile1 = open(path1, "w")

#Use csv writer
csvWriter1 = csv.writer(csvFile1)

#------------------------------------------------------------
#DECLARACIÓN DE CONTADORES
#------------------------------------------------------------

totalTweetsCounter = 0
geoTweetsCounter = 0
geoNDTweetsCounter = 0
notGeoTweetsCounter = 0


#------------------------------------------------------------
#CLASE 'LISTENER'
#------------------------------------------------------------

class TweetListener(StreamListener):

    def on_data(self,data):

        global totalTweetsCounter
        global geoTweetsCounter
        global geoNDTweetsCounter
        global notGeoTweetsCounter
        global tableName1

        #Carga cada tweet recolectado en una variable tipo json
        json_tweet_data = json.loads(data)
        totalTweetsCounter += 1

        #Coordenadas del tweet
        coords = json_tweet_data["coordinates"]

        #Fecha de creación del tweet
        tweetDate = json_tweet_data["created_at"]

        #Texto del tweet
        rawTweetText = json_tweet_data["text"].encode('utf-8')
        tweetText = rawTweetText.replace("'","''")

        #Fuente del tweet
        tweetSource = json_tweet_data["source"].encode('utf-8')
```

```
        #Idioma del texto del tweet
        tweetLang = ""
        if json_tweet_data["lang"] is not None:
            tweetLang = json_tweet_data["lang"].encode('utf-8')

        #Lugar asociado al tweet (ciudad y país)
        tweetPlace = ""
        tweetCountry = ""
        if json_tweet_data["place"] is not None:
            tweetPlace = json_tweet_data["place"][str('full_name')].encode('utf-8')
            tweetCountry = json_tweet_data["place"][str('country')].encode('utf-8')

        #-----------------------------------------------------------------------
        #BLOQUE DE CÓDIGO PARA ALMACENAR LOS TWEETS GEORREFERENCIADOS
        #-----------------------------------------------------------------------

        #Definición de texto de tweet de alta repetición
        ndTweetText = "es ahora una tendencia"
        ndIndex = tweetText.find(ndTweetText)

        if coords is not None and tweetCountry == 'Colombia':

            tweetLon = coords["coordinates"][0]
            tweetLat = coords["coordinates"][1]

            geomPointQuery = ("ST_GeomFromText('POINT("
                              + str(tweetLon) + " "
                              + str(tweetLat) + ")',4326)")

            #Definición de coordenadas de tweet de alta repetición
            ndtBool = (tweetLon == -74.0794 and tweetLat == 4.5997)

            if ndIndex == -1 and ndtBool is not True:

                geoTweetsCounter += 1
                csvWriter1.writerow([geoTweetsCounter,
                                     tweetDate,
                                     tweetText,
                                     tweetSource,
                                     tweetLang,
                                     tweetPlace,
                                     tweetLon,
                                     tweetLat])

                #Inserción de dato en la tabla
                sqlInsertIntoTable = ("INSERT INTO " + tableName1
                                      + " VALUES (" + str(geoTweetsCounter)
                                      + ",%s,'"
                                      + tweetText+ "','"
                                      + tweetSource + "','"
                                      + tweetLang+ "','"
                                      + tweetPlace + "',"
                                      + str(tweetLon) + ","
                                      + str(tweetLat)+ ","
                                      + geomPointQuery + ");")

                try:
                    cur.execute(sqlInsertIntoTable,(str(tweetDate),))
                    conn.commit()

                except Exception as e:
                    print("Hubo un error escribiendo en la tabla de tweets georreferenciados", str(e))


            elif ndIndex != -1 or ndtBool is True:
                geoNDTweetsCounter += 1

        else:
            notGeoTweetsCounter += 1

        print (tweetDate)
        print (tweetText)
        print ("Total de tweets: " + str(totalTweetsCounter))
        print ("Total de tweets georreferenciados: " + str(geoTweetsCounter))
        print ("Total de tweets no georreferenciados o de otro país: " + str(notGeoTweetsCounter))
        print ("Total de tweets georreferenciados de alta repetición: " + str(geoNDTweetsCounter))
        print ("-----------------------------------------")
        return True

    def on_error(self,status):
        print(status)
```

```
#-----------------------------------------------------------
#FUNCIÓN PRINCIPAL
#-----------------------------------------------------------
if __name__ == "__main__":

    #Creación de instancia de la clase TweetListener
    listener = TweetListener()

    #Conexión a la API de Twitter
    auth = OAuthHandler(consumer_key,consumer_secret)
    auth.set_access_token(access_token,access_token_secret)

    #Definición de "función de escucha"
    stream = Stream(auth,listener)

    #Definición de criterio de búsqueda
    bBox = [-74.286431,4.462925,-73.974975,4.829433]        #Bogotá
    #bBox = [-79.115865,-4.950340,-66.716298,12.647579]     #Colombia

    stream.filter(locations=bBox)
```