FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

MINERÍA DE TEXTO EN CIUDADES INTELIGENTES PARA LA IDENTIFICACIÓN DE PROBLEMAS EN QUITO Y LOS VALLES

AUTOR

JUAN MANUEL VIANA BARRERO

AÑO

2019

FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

MINERÍA DE TEXTO EN CIUDADES INTELIGENTES PARA LA IDENTIFICACIÓN DE PROBLEMAS EN QUITO Y LOS VALLES

Trabajo de Titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniero en Sistemas de Computación e Informática

Profesor Guía

PhD. Mario Salvador González Rodríguez

Autor

Juan Manuel Viana Barrero

AÑO

2019

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Minería de texto en ciudades inteligentes para la identificación de problemas en Quito y los Valles, a través de reuniones periódicas con el estudiante Juan Manuel Viana Barrero en el semestre 201920 orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

_____

Mario Salvador González Rodríguez

Doctor en Informática

C.C. 095837634

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Minería de texto en ciudades inteligentes para la identificación de problemas en Quito y los Valles, del estudiante Juan Manuel Viana Barrero, en el semestre 201920, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Bernarda Cecibel Sandoval Romo

Master en Ciencias de la Computación

C.C. 1709974453

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

_____

Juan Manuel Viana Barrero

C.C. 1715737118

AGRADECIMIENTOS

Primero agradezco a Dios y a mi familia, que a pesar de la distancia supieron apoyarme y guiarme durante todo este proceso, gracias por no dejar de creer en mí.

Quiero agradecer a los docentes Mario Salvador y Bernarda Sandoval por compartir sus conocimientos y orientarme en el desarrollo del presente trabajo.

A Nicole y a mi hermano Andrés, por todo la paciencia y su compañía a lo largo de este proceso.

DEDICATORIA

Quiero dedicar este trabajo a mis papás, Carlos y Patricia; ustedes han realizado un esfuerzo muy grande para darme todo lo que he necesitado a pesar de las diferentes adversidades, ustedes han permitido que yo llegue hasta aquí. Este logro es para ustedes.

RESUMEN

Las redes sociales, como Facebook, Twitter, Instagram entre otras; han revolucionado la sociedad y son un punto clave en las ciudades inteligentes, estas se han vuelto un medio para compartir experiencias y vivencias de las personas en sus entornos. En el presente trabajo, se utilizó Twitter para la recopilación de información de posibles problemas existentes en servicios prestados por el Municipio de Quito, como agua potable, luz, seguridad, movilidad, transporte público; y a través de *scripts* de Python y SQL se realiza un preprocesamiento de los datos. Esto incluye la limpieza de campos de texto para quitar espacios al inicio y final de este, la depuración de las fechas y la fuente del tuit, finalmente seleccionar el texto correcto cuando hay un retuit y evitar los tuits repetidos. Posterior a esto se realiza un proceso de minería de texto que incluye procesos de tokenización y n-gramas para poder determinar los inconvenientes y su localización. A través de una aplicación Shiny se puede visualizar por un lado el número de problemáticas que tienen los servicios (agua, luz, movilidad, seguridad y transporte público) a nivel de barrios, parroquias y zonas, por otra parte, los problemas que poseen dichos servicios. La aplicación en Shiny permite a través de la red social capturar, identificar y localizar los diferentes problemas que los servicios del Municipio presentan en percepción de la ciudadanía según sus experiencias y vivencias en su entorno.

ABSTRACT

Social networks, such as Facebook, Twitter, Instagram among others; They have revolutionized society and are a key point in smart cities, these have become in a way to share experiences of people in their environments. For the present study, Twitter has been used for information gathering about problems that exist on the services provided by the Municipality of Quito, for example, electricity, drinking water, security, mobility, public transport; and through Python and SQL *scripts* a preprocessing of the data is performed. This includes cleaning the text fields to remove spaces at the beginning and at the end of it, debugging the dates and the source of the tweet, to finally select the right text when there repeated text. Subsequently, the use of text mining techniques such as tokenization and n-gramas is used to determine the problems and their location. Through a Shiny application you can visualize on the one hand the number of problems that services have (water, electricity, mobility, security and public transport) at the level of neighborhoods, parishes and areas, on the other hand, the problems that these services have. The application in Shiny allows through the social network to capture, identify and locate the different problems that arise in citizens according to their experiences in their environment.

ÍNDICE

1. INTRODUCCIÓN 1

1.1. Antecedentes 1

1.2. Alcance 2

1.3. Justificación 3

1.4. Objetivos 4

1.4.1. Objetivo General 4

1.4.2. Objetivos Específicos 4

2. MARCO TEORICO 5

2.1. Minería de Datos 5

2.1.1. Recolección de Información 5

2.1.2. Preprocesamiento 6

2.1.3. Minería o Procesamiento 6

2.1.3.1. Limpieza de palabras y *Stop Words* 7

2.1.3.2. Tokenización 7

2.1.3.3. N-Gramas 7

2.1.4. Análisis a través de herramientas de visualización de datos 8

3. DESARROLLO 8

3.1. Recolección de Información 8

3.2. Preprocesamiento 10

3.3. Minería o Procesamiento 15

3.3.1. Limpieza de palabras y *Stop Words* 15

3.3.2. Tokenización 18

3.3.3. N-gramas 19

3.4. Análisis a través de herramientas de visualización de datos 20

4. ANÁLISIS DE RESULTADOS 25

4.1. Generalidades 25

4.2. Problemáticas Generales 29

4.2.1. Agua 30

4.2.2. Energía Eléctrica 31

4.2.3. Movilidad 32

4.2.4. Seguridad 34

4.2.5. Transporte Público 34

4.3. Mapa de Problemáticas 36

4.4. Red de Palabras 39

4.4.1. Agua 40
4.4.2. Energía Eléctrica 40
4.4.3. Movilidad 41
4.4.4. Seguridad 42
4.4.5. Transporte 43

5. CONCLUSIONES Y RECOMENDACIONES 45

5.1. Conclusiones 45

5.2. Recomendaciones 45

REFERENCIAS 47

ANEXOS 49

1. INTRODUCCIÓN

1.1. Antecedentes

Smartcities es un término que se ha empezado a utilizar en los últimos años, debido a la necesidad que existe de transformar a las ciudades actuales en lugares que administren sus recursos y servicios de manera más eficiente. Así, a través de sensores o controles digitales recopilar información en tiempo real para generar una amplia cantidad de datos, que permitan tomar mejores decisiones y tener mejores resultados para las ciudades y sus habitantes (Harmon & Lee, 2016).

El termino de *Smartcites* no es únicamente el introducir sensores, o capturar información acerca de la ciudad. También, es tener una orientación más humana y previamente a la implementación de tecnologías definir las necesidades, deseos e intereses de los ciudadanos. Estas herramientas se han convertido en una ayuda para los gobiernos y sus interesados, logrando identificar necesidades de los ciudadanos y disminuir la falta de confianza entre ellos (Almeida, Moreira, & Danilo, 2018).

Dentro de las ciudades inteligentes está el uso de las redes sociales para poder analizar las opiniones de los ciudadanos y las reacciones que pueden llegar a tener ante gobernantes o a entidades gubernamentales. En los últimos años redes sociales como Twitter han brindado la oportunidad de realizar análisis sobre las interacciones que tienen los usuarios dentro de la red social, por ejemplo, a través de funcionalidades como *retweet* (RT) o en español retuitear; donde la plataforma lo define como: “Un tuit que compartes públicamente con tus seguidores. (…) puedes retuitear o retuitear con comentario desde tu propio tuit”(Twitter, 2018). Se puede determinar opiniones que existen sobre un tema específico o sobre información que es publicada por entidades públicas o sus miembros (Garimella, Weber, & Tech, 2016). Otra herramienta que brinda Twitter

para analizar el comportamiento, necesidades u opiniones de los usuarios es a través del uso de menciones (#) donde se generan tendencias en la red social y se puede conocer problemas actuales dentro de un barrio o ciudad (Garimella, Mathioudakis, Gionis, & Morales, 2016).

La red social ha permitido generar estudios de análisis de datos, otorgando la oportunidad de detectar inconvenientes como en el sector de la movilización. Todo esto a través de las APIs que la plataforma tiene disponibles, en este caso para el consumo de información de tuits donde se obtienen datos como localización, texto, fecha de creación etc. En Londres se ha realizado la implementación de una herramienta que recopila procesa y analiza la información del tráfico de la ciudad. La herramienta tiene como finalidad identificar dentro del tráfico problemas, tendencias y comportamientos en la ciudad. Esta ha llegado al punto donde se ha realizado una predicción de aglomeraciones o atascos por eventos no planificados. Este trabajo fue realizado con el uso de dos herramientas, la primera, Spark SQL, la cual procesaba, clasificaba y resumía la información, y la segunda, Tableau, que con la información obtenida construía los gráficos (Suma, Mehmood, & Albeshri, 2018).

El presente trabajo pretende identificar por una parte el número de inconvenientes que tienen los servicios (agua, luz, movilidad, seguridad y transporte público) a nivel de barios, parroquias y zonas, por otra parte, los problemas que poseen dichos servicios. Para lograr esto se utilizan dos herramientas, una de ellas se encarga de la recopilación de tuits semanalmente y realizar un procesamiento de lenguaje natural y obtener datos que puedan ser utilizados por la segunda herramienta, que permite mostrar de forma gráfica toda la información relacionada al número de inconvenientes de los diferentes servicios básicos y los problemas que pueden estar tomando lugar.

1.2. Alcance

La propuesta de este proyecto es implementar una solución que a través del uso

de APIs de Twitter recopile información sobre tuits realizados en la ciudad de Quito que tengan una relación directa sobre los servicios que presta el Municipio, como, por ejemplo: transporte público, movilización, empresa eléctrica, etc. Esta búsqueda viene acompañada de un proceso de filtrado previo que permita descartar tuits que no tenga relación con el objetivo, como por ejemplo que no se esté mencionando un problema sobre el servicio.

Con la información recopilada se realiza un preprocesamiento, donde utilizando *scripts* en Python y SQL, se clasifica la información obtenida, luego se realizan procesos de limpieza de texto, depuración de fechas y fuente de tuits, por último, se obtiene el texto para tuits y retuits y se eliminan los duplicados. Con la finalidad de tener la información depurada para realizar procesos de minería de texto.

Una vez la información sea procesada, se utilizan herramientas de minería de texto como tokenización y bigramas, la primera herramienta permitirá identificar el número de veces que un barrio, parroquia y zona es mencionada en los tuits, y la otra, el n-grama de dos palabras, permitirá la detección de los posibles problemas que experimenta cada uno de los servicios en cuestión.

Después de realizar la minería de texto se busca implementar gráficos que den la información necesaria para reconocer donde se encuentran la mayor cantidad de problemas en el servicio prestado por el Municipio, y cuales podría ser las problemáticas que tiene el servicio, y a través de mapas o gráficas se propone generar un método ilustrativo de resultados que sea claro y entendible.

1.3. Justificación

En el Ecuador el uso de las redes sociales en dispositivos móviles a aumentado con el paso de los años, especialmente en las ciudades donde el 92.4% de los ciudadanos que poseen celular utilizan las redes sociales (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2015).

Actualmente diferentes entidades municipales cuentan con una cuenta de Twitter, donde no solo se pública información con respecto a los servicios prestados por las empresas, también los usuarios las utilizan como medio para reportar incidentes dentro de los servicios o notificar sobre algún problema existente, se ha convertido en una manera de hacer un reclamo por parte de la ciudanía hacia las entidades municipales.

Esto nos lleva a poder generar una aplicación que se pueda convertir en una herramienta para las entidades municipales donde a través de gráficos identificar los lugares más conflictivos, lo que podría ayudar en la toma de decisiones y construcción de políticas públicas que reduzcan el número de problemas en la ciudad, otorgando al ciudadano una sensación de mejora en los servicios.

1.4. Objetivos

1.4.1. Objetivo General

Implementar una aplicación en Shiny que a través de la recopilación de tuits y minería de texto permita el reconocimiento y mapeo de inconvenientes dentro de los servicios prestados por el Municipio de Quito.

1.4.2. Objetivos Específicos

Realizar la recopilación y preprocesamiento de los tuits con información de posibles problemas existentes en servicios prestados por el Municipio de Quito como: agua, electricidad, movilidad, transporte público y seguridad.

Procesar los tuits recopilados con técnicas de minería de texto para poder determinar la cantidad de inconvenientes por barrio, parroquia y zona municipal e identificar cuáles son estos problemas.

Elaborar gráficos y mapas de los inconvenientes y sus ubicaciones a través de una aplicación Shiny

2. MARCO TEORICO

2.1. Minería de Datos

La minería de datos es un tema que ha ido ganando espacio a medida que los volúmenes de información crecen cada día ya que superan cualquier capacidad humana para poderlos procesar. Está definida como la ciencia de la extracción de conocimiento útil de grandes repositorios de datos, y puede ser considerado como una aplicación orientada a diferentes campos multidisciplinarios. Por ejemplo, se encuentra relacionado con la lingüística computacional, procesamiento del lenguaje y la recuperación de información, además de contribuciones de tipo estadístico y *machine learning*. Hoy en día las personas tienen un gran acceso a cualquier tipo de información de forma ilimitada en formato digital de texto, para esto la minería de texto se presenta como una herramienta muy útil para aplicaciones de investigación (E. Banchs, 2013).

Dentro de un proceso de minería de datos se pueden encontrar los siguientes pasos: recolección de información, preprocesamiento, procesamiento o minería y análisis a través de herramientas de visualización de datos

2.1.1. Recolección de Información

La recopilación de información es el primer paso, y se define como el proceso de acceder a la información, la misma que puede provenir de diferentes fuentes, por ejemplo: correos, documentos, bases de datos, etc. (*OPENMINTED COMMUNICATIONS, 2018)*. Una vez seleccionada la fuente es necesario identificar como se va a acceder a la misma, por ejemplo: en el caso de ser una base de datos interna, se puede utilizar una consulta SQL.

Para acceder a la información provista por una empresa, en la mayoría de los casos es necesario establecer el método de conexión, para esto suelen existir medidas de seguridad o autenticación. En el caso de Twitter es necesario solicitar un token para acceder a la información. Este token puede ser utilizado por las diferentes herramientas disponibles para realizar las consultas. Una de ellas es a través de la librería *rtweet* de R, la cual por medio de una aplicación de Twitter y un *script* de R realiza el vínculo de la conexión (Michael W. Kearney, 2019) . Otra herramienta es la librería *tweepy* para Python la cual por medio de un *script* permite establecer la conexión con Twitter (Roesslein, 2019).

2.1.2. Preprocesamiento

El preprocesamiento o preparación y transformación de datos, consiste en preparar la información e identificar las características representativas. (*OPENMINTED COMMUNICATIONS*, 2018). Es importante recalcar que este paso debe dejar la información lista para el proceso de minería.

Las transformaciones son realizas para darle un formato deseado a uno o varios datos recopilados, por ejemplo, transformar una fecha para obtener el nombre del día, o transformar un campo de tipo texto a numérico. Esto puede ser realizado en diferentes lenguajes, por ejemplo: Python o SQL, aunque para este último la información debería estar almacenada en una base de datos.

2.1.3. Minería o Procesamiento

Este paso dependerá de cómo la información sea capturada. Por ejemplo, esto puede ser realizado a nivel de base de datos, este proceso consiste en realizar cubos de información a través de las relaciones existentes en la base de datos. (Microsoft, 2019b). Pero en el caso en que la minería trabaje con texto o lenguaje natural, se realiza un proceso de procesamiento de lenguaje natural que incluye lo siguiente: limpieza de palabras y *stop words*, tokenización y n-gramas.

Este proceso se puede realizar con diferentes herramientas. En el mercado actualmente existen herramientas como Orange, una herramienta libre y fácil de usar que provee una interfaz de programación visual y además, incluye visualización interactiva de datos (Demsar J y otros, 2013). Por otro lado se puede utilizar R, un software libre que sirve para computación de gráficos y computación estadística (R Foundation, 2019). R tiene un mecanismo de búsqueda para sus archivos, generalizan los términos disponibles más allá de los alias e introduce un poco de flexibilidad en la búsqueda, todo esto lo realiza a través de *scripts*, los mismos que deben ser programados (Chambers & Hand, 2008).

2.1.3.1. Limpieza de palabras y *Stop Words*

Al extraer texto de una fuente de internet es muy común encontrar dentro del texto palabras que pueden generar ruido en un análisis. Para reducir este ruido se puede eliminar una lista de palabras no deseadas. Esta lista puede ser generada o puede estar predefinida cómo las palabras conocidas como *stop words*, que son aquellas que expresan una relación gramatical y están establecidas para cada idioma (Beysolow II, 2018).

2.1.3.2. Tokenización

La tokenización consiste en separar palabras o grupos de palabras en una oración o en un texto (Beysolow II, 2018) . Está herramienta permite que se pueda comparar un texto con respecto a una lista de palabras para conocer, por ejemplo, la frecuencia del texto tokenizado.

2.1.3.3. N-Gramas

Un N-Grama se puede entender como la secuencia de N palabras, como por

ejemplo un trigrama donde se obtienen secuencias de tres palabras. Entonces, lo que se busca es la secuencia de una palabra con las anteriores con la finalidad de obtener la frecuencia de esa secuencia y conocer el número de veces que ese grupo de palabras se repiten a lo largo de un texto.(Kapadia, 2019).

2.1.4. Análisis a través de herramientas de visualización de datos

La minería de datos genera resultados crudos, los mismos que deben ser visualizados para poder ser interpretados (*OPENMINTED COMMUNICATIONS*, 2018). En el mercado existen herramientas tanto libres como pagadas que brindan una serie de beneficios en la visualización de los datos. Una de ellas es Power BI, una plataforma paga que permite conectarse con múltiples fuentes de datos y crear visualizaciones e informes, todo esto a través de una misma interfaz (Microsoft, 2019a). Por otro lado, Shiny es una librería de R, por ende, gratuita, que permite la creación de aplicaciones web directamente desde R, estas aplicaciones pueden ser complementadas a través de acciones *JavaScript* o temas CSS, además esta aplicación puede ser publicada utilizando los servidores Shiny, donde existen planes tanto gratuitos como pagos (R Foundation, 2017) .

3. DESARROLLO

El presente proyecto se encuentra dentro del campo de la minería de datos. Con el objetivo de implementar una aplicación web en Shiny que permita a través de la red social capturar, identificar y localizar los diferentes problemas que los servicios de agua potable, luz eléctrica, movilidad, seguridad y transporte público presentan en el Municipio de Quito.

3.1. Recolección de Información

Como fuente de datos se establece obtener los tuits sobre los servicios

municipales, para esto es necesario implementar una conexión con el servicio web de Twitter. El proceso consiste en una serie de pasos, primero, con cualquier cuenta de Twitter se debe aplicar para obtener una cuenta de desarrollador, segundo, con la cuenta de desarrollador se debe crear una suscripción o un ambiente de desarrollo, en este punto Twitter solicitará el motivo por el cual se desea acceder a la información, finalmente la red social verifica y autoriza la creación del ambiente. Este ambiente generado cuenta con *tokens* de seguridad, los que deben ser utilizados para establecer la conexión con Twitter.

Cuando Twitter otorga el acceso como cuenta de desarrollador establece un perfil de cuenta gratuita, esto genera ciertas limitantes para acceder a la información, entre ellas, solo se puede acceder a tuits generados en los últimos 15 días y el número de tuits diarios a recopilar es limitado. Para corregir esto fue necesario implementar recopilaciones de información semanales e incluir validaciones para detener la búsqueda en caso de superar la cuota diaria con el fin de evitar que la conexión sea bloqueada.

Para realizar la búsqueda de tuis se utilizó la librería *tweepy* de Python, esto debido que a que existe gran cantidad de documentación disponible y hay una facilidad de codificación con respecto a otras herramientas del mercado. Para utilizar la librería fue necesaria la configuración de los siguientes parámetros: token de seguridad, el texto que este está buscando, el idioma, la fecha inicial, y el modo del tuit. Es importante recalcar que el modo de tuit usado es *“extended”* dado que permite obtener el cuerpo del tuit completo (Twitter, 2017). Twitter actualmente permite ingresar tuits con 280 caracteres, sin embargo, si no se coloca el modo extended en la búsqueda, el texto del tuit recopilado solo contendría los primeros 140 caracteres.

Una vez realizada la búsqueda, el API de Twitter retorna un archivo JSON, este contiene dos principales secciones, la primera es con respecto al usuario y la otra tiene información del tuit. El formato se puede ver en el ANEXO 1

Se procede a obtener los tuits para los cinco servicios: agua potable, energía eléctrica, movilidad, seguridad y transporte público. Para cada uno de ellos se realizan múltiples búsquedas con diferentes menciones, palabras o etiquetas.

Las menciones utilizadas son las cuentas oficiales de cada uno de los servicios, @aguadequito, @ElectricaQuito, @AMTQuito, @PoliciaEcuador, @TransporteQuito. En el caso de las etiquetas, se consultó las cuentas de Twitter de cada uno de los servicios para conocer cuáles eran utilizadas para comunicar o notificar un inconveniente, por ejemplo: #AguaDeQuito, #EEQ o #AMTInforma.

En el siguiente fragmente de código se puede apreciar un ejemplo, en el cual se está buscando tuits que hagan mención a la cuenta oficial de la empresa “Agua de Quito” y tenga la palabra “sin” dentro del texto, que el idioma sea en español, que busque desde el 5 de mayo del 2019 y el modo del tweet sea extendido.

tweepy.Cursor(api.search, q= @aguadequito+sin', lang="es”, since=2019-05-15, tweet_mode= "extended")

Es importante recalcar que para algunos servicios la búsqueda incluyo algunos parámetros adicionales, en el caso de la movilización el texto de búsqueda utiliza la cuenta oficial de la entidad municipal seguida de una cara triste (☹). Este operador de búsqueda que provee Twitter filtra mensajes con actitud negativa (Twitter, 2019). Por otro lado, con respeto al transporte público se busca la cuenta oficial de la empresa municipal y que contenga alguna de las siguientes palabras: corredor sur occidental, ecovia, trole, buses o bus. También, se incluye otra búsqueda que tenga la cuenta oficial de la AMT seguido por alguna de las siguientes palabras: buses o bus.

3.2. Preprocesamiento

En el preprocesamiento se debe seleccionar la información que se desea

almacenar en la base de datos. En este caso, la información obtenida es filtrada y almacenada en una base de datos Microsoft SQL 2016 a través de un *script* de Python, una vez almacenada se utilizan *scripts* SQL para realizar las diferentes transformaciones. Se realizó de esta manera dado que la recopilación inicial fue realizada en Python y se combinó con *scripts* SQL debido a la familiaridad con las funciones y el proceso de transformación que se puede realizar con esta herramienta.

Primero se realiza un *script* en Python el cual toma el archivo JSON que la API retorna como resultado, en caso de que existan varios archivos de búsqueda por servicio el *script* se encarga de unirlos. Después de tener al archivo listo se seleccionan las columnas que se van a almacenar para conservar únicamente los datos necesarios, con respecto a los campos del usuario se van a almacenar las siguientes columnas: *id, name, screen_name, location, friends_count, statuses_count, verified, url.* Por el otro lado, los campos a almacenar del tuit son: *created_at, id, id_str, texto, retweet_count, favorite_count, retweeted truncated, source, lang*.

Además, se realizan procesos adicionales con las columnas, por empezar se buscan los tuits que fueron retuiteados, esto se logra construyendo una columna con las tres primeras letras del texto, si la columna contiene las letras ‘RT’ este es considerado como un retuit y en una nueva columna se almacena el resultado si fue verdadero o falso. También, la columna *text* que contiene el texto es almacenada haciendo una conversión a Unicode. Finalmente, dentro del archivo se encuentran columnas con listas para los siguientes elementos del tuit: menciones, etiquetas, direcciones url y archivos multimedia; para cada uno de ellos se agrega una columna nueva con el número de elementos de la lista y se crea una cadena con todos los elementos que componían la misma.

Una vez realizadas estas transformaciones por el *script* de Python, se utiliza la librería SQL para establecer la conexión con una base de datos Microsoft SQL Server 2016, dentro de la misma se generó previamente una tabla para cada uno

de los servicios, cada una contiene la misma estructura, esta se puede observar en la Figura 1, donde se ilustra los campos de la tabla “aguapotable”

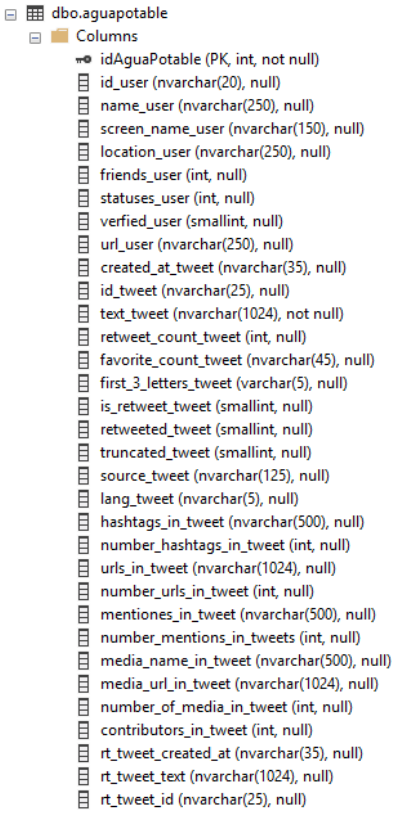

Figura 1. Campos de la tabla “aguapotable” de la base de datos

Después de que los tuits son almacenados se generan *scripts* SQL para realizar transformaciones a los diferentes campos. Por empezar la fecha que retorna Twitter está en la zona horario cero, en el siguiente formato “*Mon Jan 07 23:55:51 +0000 2019*”. Para poder estandarizar, primero es necesario restar 5 horas para estar en la misma zona horaria correspondiente a la ciudad de Quito, luego a través de la ubicación de los diferentes caracteres se reconoce el día, mes, año y hora. Finalmente, con la combinación de estos campos se construye la fecha y se agregan columnas adicionales para conocer el nombre del día de la semana y del mes. Para poder realizar todo este proceso se codifica un *script* SQL, este se puede ver en el ANEXO 2, el mismo que al ser ejecutado retorna el resultado que se puede observar en la Tabla 1, aquí podemos identificar que las fecha son desglosadas en diferentes campos, todo esto con la finalidad de poder realizar

más adelante un análisis más detallado. También, cabe recalcar que se conserva el id del tuit dado que este campo permitirá unir la fecha construida con el tuit.

Tabla 1.
Procesamiento Fechas

TUIT ID	FECHA	DIA	MES	ANIO	NOM DIA	NOM MES	HORA
1112794289145307139	2019-04-01 14:10:07.000	1	4	2019	*Monday*	*April*	14
1112813816843259905	2019-04-01 15:27:43.000	1	4	2019	*Monday*	*April*	15
1112827192713662464	2019-04-01 16:20:52.000	1	4	2019	*Monday*	*April*	16
1112915918730952705	2019-04-01 22:13:26.000	1	4	2019	*Monday*	*April*	22
1113047886307233792	2019-04-02 06:57:50.000	2	4	2019	*Tuesday*	*April*	6
1113048861751341057	2019-04-02 07:01:42.000	2	4	2019	*Tuesday*	*April*	7
1113085908188889088	2019-04-02 09:28:55.000	2	4	2019	*Tuesday*	*April*	9
1113097424686415872	2019-04-02 10:14:41.000	2	4	2019	*Tuesday*	*April*	10
1113228882218557440	2019-04-02 18:57:02.000	2	4	2019	*Tuesday*	*April*	18
1113234155343613952	2019-04-02 19:18:00.000	2	4	2019	*Tuesday*	*April*	19

Nota. 10 registros almacenado en la base de datos, tabla aguapotable. NOM DIA = Nombre Día, NOM MES = Nombre Mes

Después, se procede a eliminar espacios en blanco que pudieran existir en la parte posterior o anterior al texto. Utilizando las funciones de *LTRIM* y *RTRIM* que ofrece SQL se eliminan estos espacios.

Otra columna a la que se le realiza una transformación es la fuente, originalmente Twitter retorna la columna con el siguiente formato: *“Twitter for Android”*

Dentro de la etiqueta <a> se encuentra el nombre de la fuente. Para obtenerlo es necesario clasificar y transformar en un nombre estándar. El proceso consiste en clasificar las diferentes fuentes y otorgarles un nombre, como se muestra en

la Tabla 2 donde a cada etiqueta se la puede conocer por el nombre asignado. En la Tabla 2 se visualiza el resultado de esta transformación. Como por ejemplo para la etiqueta anterior el nombre es “ANDROID”.

Tabla 2.
Procesamiento de Fuentes

SOURCE_TUIT	SOURCE_TUIT CLASIFICADO
Twitter for iPhone	*IPHONE*
Twitter for iPad	*IPAD*
Twitter for Android	*ANDROID*
Twitter Lite	*TWITTER_LITE*
Mobile Web (M2)	*WEB_MOBILE*
Twitter Web App	*TWITTER_WEB_APP*
Pypbot	*PYPBOT*
Echofon	*ECHOFON*
Hootsuite Inc.	*HOOTSUITE*
Publicaciones desde yo veo veo	*OTRO*
TuitDeck	*TWITTER_DECK_APP*
Twitter Web Client	*TWITTER_WEB_CLIENT*

Nota. SOURCE_TUIT = Texto sin transformar, SOURCE_TUIT CLASIFICADO = Texto transformado

Una vez se completan todas las transformaciones, ahora es necesario seleccionar el texto correcto de un tuit cuando este fue retuiteado, esto se debe que al realizar un retuit el texto original es almacenado en una columna diferente. Después de seleccionar la columna de texto correcta para cada uno de los casos, como último paso, se filtran los tuits repetidos para evitar información duplicada, esto por medio del identificador único del tuit.

Todo el proceso de transformaciones puede ser apreciada en el ANEXO 3, donde se encuentra el *script* SQL encargado de generar una vista para cada servicio que contiene todas las transformaciones realizadas. Los campos que contiene esta vista pueden se puede observar en la Figura 2, donde se despliega la estructura de la vista para el servicio de agua potable.

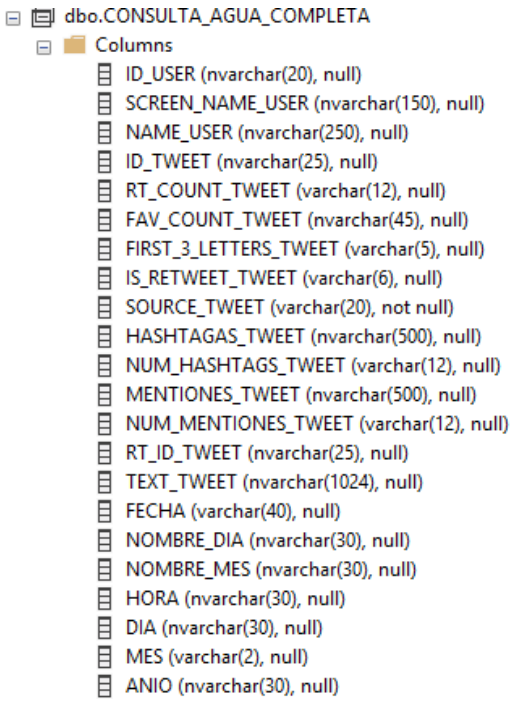

Figura 2. Campos de la vista CONSULTA_AGUA_COMPLETA

Una vez realizada la vista para cada uno de los servicios, los datos se exportan a un archivo de texto plano delimitado por tabulaciones. Estos van a hacer los archivos que los *scripts* de minería utilicen como fuente de datos.

3.3. Minería o Procesamiento

La herramienta seleccionada para este proceso fue R, debido a la versatilidad que brinda para poder codificar los *scripts*.

3.3.1. Limpieza de palabras y *Stop Words*

El primer proceso a realizar es la limpieza de texto, esto a través de un *script* de R, que se encarga de varios procesos con el texto, entre ellos eliminar palabras innecesarias o que pueden influir en la frecuencia de palabras y no tienen ninguna relevancia para la detección de problemas, además de

transformaciones al texto como tal para poder estandarizarlo.

Lo primero es tener el texto del tuit en minúsculas, esto permite tener todo el texto estandarizado. Después es necesario eliminar caracteres especiales dentro del texto, tales como: @, /, |, etc. También, se debe eliminar cualquier enlace web insertado dentro del texto. Luego hay que quitar cualquier tipo de acentuación o puntuación que contenga el texto del tuit, y finalmente es necesario intercambiar la letra “ñ” por “n”. En la Tabla 3 se observa en gran parte las transformaciones mencionadas. Todas estas son realizadas con la finalidad de hacer comparaciones con otras palabras y encontrar coincidencias.

Tabla 3.
Transformación de palabras

ANTES	DESPUÉS
Carcelén	carcelen
Amaguaña	amaguana
Itchimbía	itchimbia
Iñaquito	inaquito

Después de realizar estos cambios, es necesario eliminar ciertas palabras del texto. Entre ellas están palabras como cuentas de Twitter, por ejemplo: teleamazonasec, mauriciorodasec, bomberosquito, lorohomero; además, se quitan pronombres, conjunciones, etc. Esto se realiza retirando palabras conocidas como *stop words*, estas son palabras que se utilizan para hacer pausas o uniones en el texto, En la tabla 4 se observan algunas de estas palabras que son removidas con el *script*. Sin embargo, esto únicamente se utiliza en el proceso de tokenización donde se busca encontrar coincidencias con barrios, parroquias y zonas. Para el proceso de n-gramas, como se busca la relación entre palabras, se eliminan solo algunas de estas, como, por ejemplo: el, la, las, los, ya, su, al, etc. y no se utiliza la lista completa de *stop words* que contiene más de 300 palabras.

Tabla 4.
Palabras vacías o stop words

stopwords(“*spansish*”)
de
la
que
el
en
y
a
los
del
se

Nota. Lista stop words. Se muestran las primeras 10 de 308 palabras

Una vez realizada la limpieza de los tuits también se debe hacer los mismo para el caso de nombres de parroquias, barrios y zonas que se obtienen de los archivos publicados por la ciudad de Quito (Secretaría de Territorio Hábitat y Vivienda, 2017). Cada una de las bases recopiladas para barrios, parroquias y zonas tienen un archivo de tipo *shapefile*. Aquí es necesario utilizar el programa ArcMap 10.5 para exportar cada uno de ellos a un archivo xls. El mismo que es utilizado por el *script* en R para realizar las siguientes transformaciones adicionales, como, por ejemplo: cada vez que se encuentre los caracteres “s.” es la abreviación de san, los caracteres “col.” es la abreviación para colegio, entre otras. Estas transformaciones se pueden observar en la Tabla 5.

Tabla 5.
Transformaciones para nombres de barrios, parroquias y zonas

ANTES	DESPUÉS
s.	san
col.	colegio
sta.	santa
coop.	cooperativa

3.3.2. Tokenización

Este proceso busca obtener el número de veces que una palabra se repite dentro del texto. Esto permite poder reconocer las palabras más usadas y así llegar a barrios o problemáticas existentes. Este proceso se realiza después de que tanto el texto del tuit como los nombres de barrios, parroquias y zonas han pasado por el proceso de limpieza y depuración.

El conteo de coincidencias es realizado para cada uno de los niveles de geolocalización. Primero se realiza a nivel de barrios, luego a nivel de parroquias y finalmente a nivel de zonas. Para este último se estableció un procedimiento distinto, primero se debe realizar una comparación a nivel de parroquias y luego se resume por zona. Una vez se realiza este proceso se agregan las columnas de geolocalización provenientes del archivo de zonas obtenido del Municipio.

El fragmento de código ubicado en el ANEXO 4 realiza los siguientes pasos, primero convierte en un *dataframe* todos los nombres de los barrios, luego realiza una comparación con respecto a ‘d’ una lista de las palabras de los tuits. Después se procede a estructurar la información, eliminar resultados sin coincidencias y ordenar por la frecuencia.

Para encontrar coincidencias a nivel de las paradas del transporte público se separa el proceso en dos partes. La primera con paradas del sistema integrado, lo que compone a los diferentes corredores de transporte como: Ecovia, Trole, Corredor-Suroccidental, etc. La segunda parte comprende todo el resto de las paradas ubicadas en las calles del distrito.

Para la primera parte, se utiliza el recurso disponible en línea por parte de la alcaldía para las paradas del sistema integrado de transporte público (Distrito Metropolitano de Quito, 2016). Una vez importada la información a través del *script* se utiliza el nombre de la parada y también se realiza un resumen por el corredor. En el otro caso, se utiliza los archivos geográficos disponibles para las

paradas de buses del transporte público (Distrito Metropolitano de Quito, 2015). Después de importarlos se utiliza el campo que indica la calle principal de la parada para encontrar coincidencias dentro del texto. El fragmento de código se puede observar en el ANEXO 5

Otro caso adicional en el que la tokenización difiere, es con respecto a la comparación de ejes viales, para esto es necesario utilizar la fuente de vías provista por el gobierno local (Distrito Metropolitano de Quito, 2018). La fuente es modificada utilizando el programa ArcMap 10.5 donde se realiza la una unión de calles con la finalidad de evitar que estas se repitan. Luego de haber concluido este proceso el archivo xls es generado e importado. El *script* realiza una búsqueda de estas calles en los tuis para obtener la frecuencia, cabe aclarar, que en este caso no se toma en cuenta la ubicación geográfica de las calles.

Una vez concluidos todos los procesos se genera un archivo de tipo rda para cada uno de los servicios. Dentro de estos están almacenados *dataframes* que contiene la información para generar los gráficos de la aplicación

3.3.3. N-gramas

En el presente trabajo se realizan bigramas, para esto el *script* genera una lista de palabras con las cuales va a realizar las comparaciones, la misma que proviene de los tuits y se encuentra procesada, es decir, se quitaron palabras no deseadas como conjunciones, caracteres especiales, puntuación, etc.

Luego de obtener esta lista de palabras, el *script* busca la relación entre cada una de estas con respecto al tuit y obtiene el peso cada relación, esto se puede observar en la Tabla 6, donde, por ejemplo, se muestra que la palabra “más” va acompañada de “información” y tiene una frecuencia o peso de 82.

Tabla 6.
Resultado n-gramas

PALABRA 1	PALABRA 2	FRECUENCIA
mas	informacion	82
suspension	agua	82
cambio	agua	80

Después de concluir el proceso se genera un archivo de tipo rda para cada uno de los servicios. Dentro de estos están almacenados *dataframes* que contiene la información para generar los gráficos de la aplicación

3.4. Análisis a través de herramientas de visualización de datos

La aplicación encargada para visualizar los datos fue construida utilizando una librería de R, Shiny. Fue la herramienta escogida debido a que el procesamiento fue realizado en R y provee de varias facilidades al momento de programar.

Para construir una aplicación utilizando esta librería, es necesario entender las secciones en que esta se divide. Primero está la parte que se va a ejecutar en el servidor, en esta sección se coloca el código encargado de leer el archivo necesario y generar el gráfico, por otra parte, se encuentra la interfaz gráfica, aquí se debe incluir todos los componentes visuales que va a tener la aplicación, por ejemplo, cuadros de texto, gráficos, títulos, etc. Además, existe una sección opcional que puede ser incluida que consiste en generalidades, aquí se pueden colocar listas o variables generales para que puedan ser utilizadas por cualquier sección de la aplicación.

La aplicación al costado izquierdo cuenta con un menú, como se puede observar en la Figura 3, existen dos secciones principales, *Dashboard* y Problemáticas. Dentro de Problemáticas se puede acceder a tres sub opciones: Generalidades, *Map* y Palabras. Finalmente, en la parte inferior, el menú va a permitir escoger que tema se desea ver dentro de cada opción del menú, así la información

visualizada puede ser de agua, luz, movilidad, seguridad, transporte público o todos los servicios.

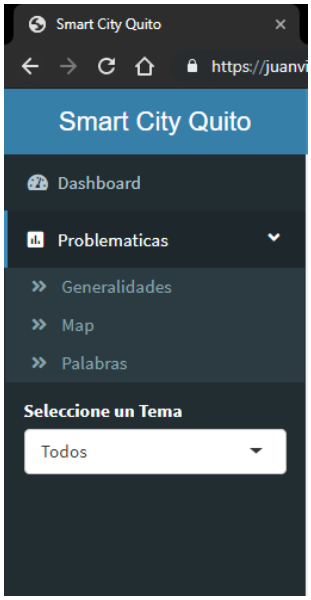

Figura 3. Menú de la aplicación

Dentro de la sección *Dashboard*, se puede apreciar tres secciones internas. La primera en la parte superior como se muestra en la Figura 4, esta consta de un resumen del conteo de tuits y el número de usuario únicos que publicaron tuits.

Figura 4. Parte superior Dashboard

En la sección media superior, como se observa en la Figura 5, hay gráficos con respecto a la distribución de tuits en el tiempo, de izquierda a derecha se ve el número de tuits por mes, por semana y finalmente por día de la semana.

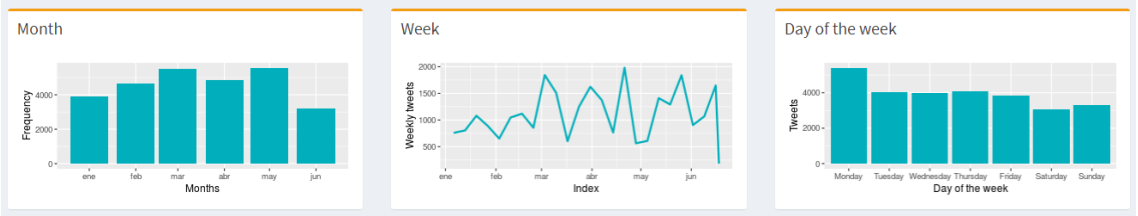

Figura 5. Parte media superior *Dashboard*

Continuado con el *Dashboard* del aplicativo, en la sección media inferior como se muestra en la Figura 6, el graficó del costado izquierdo muestra el número de tuits diarios, además este gráfico tiene una opción donde permite seleccionar el tipo de gráfico con el que se quiere observar, de líneas o de barras. En el otro lado está el conteo de menciones y hashtags en los tuits.

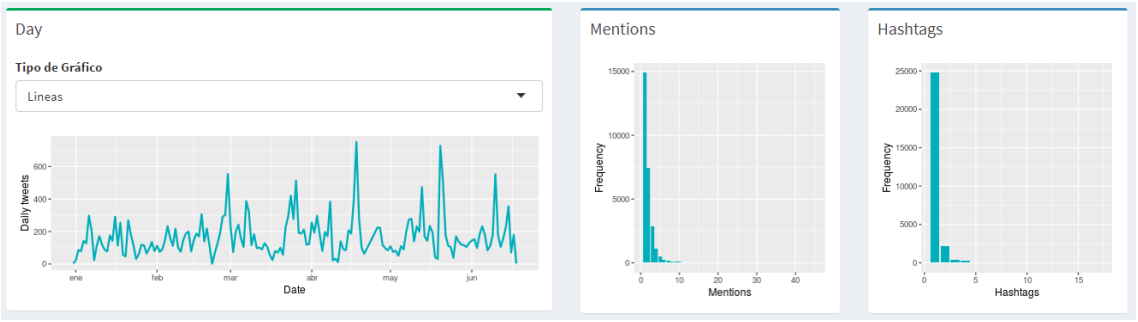

Figura 6. Parte media inferior *Dashboard*

Finalmente, el *Dashboard* cuenta con una sección inferior, como se ve en la Figura 7, hay dos gráficos, el que está al lado izquierdo, muestra un conteo del número de tuits por fuente, y al otro lado un gráfico de barras que permite visualizar el número de tuits para cada hora.

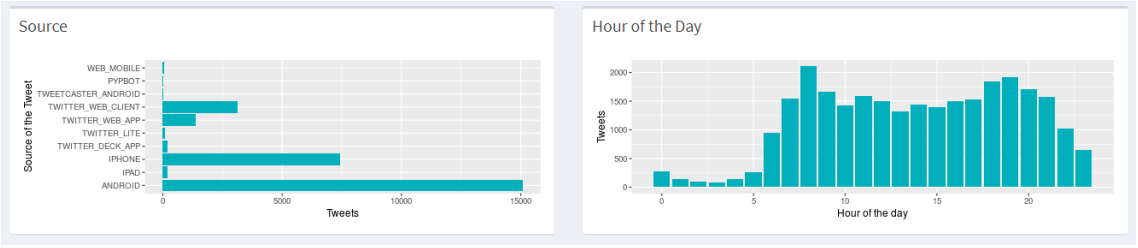

Figura 7. Parte inferior Dashboard

Por otro lado, la parte de Problemáticas se divide en 3 secciones. La primera Generalidades, tiente dos secciones, como se ve en la Figura 8, la gráfica que puede ser de barras o líneas muestra el conteo de pablas de los tuits y se encuentra ubica en la parte superior.

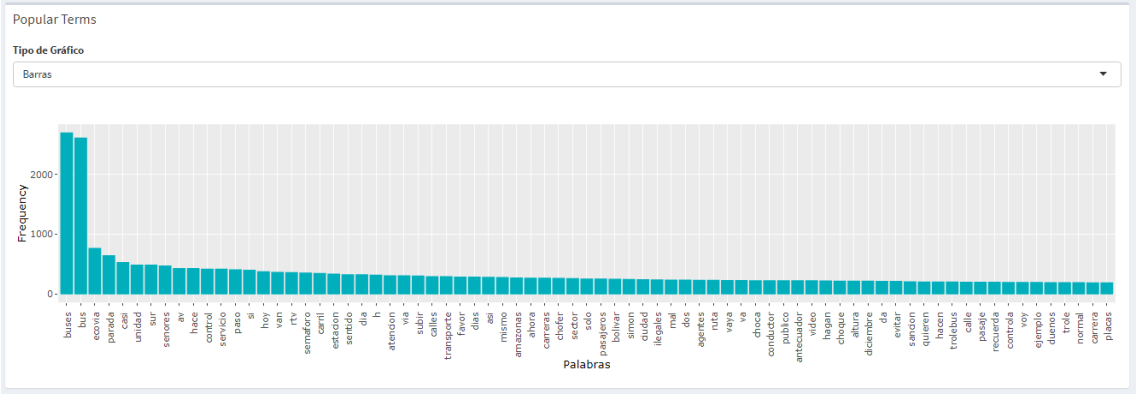

Figura 8. Parte superior Problemáticas - Generalidades

En la parte inferior se muestra el conteo de barrios, parroquias y zonas, como se observar en la Figura 9, además en el costado izquierdo, aparecerá el conteo de paradas o vías en vez de barrios cuando selecciona la opción de transporte público o movilidad respectivamente.

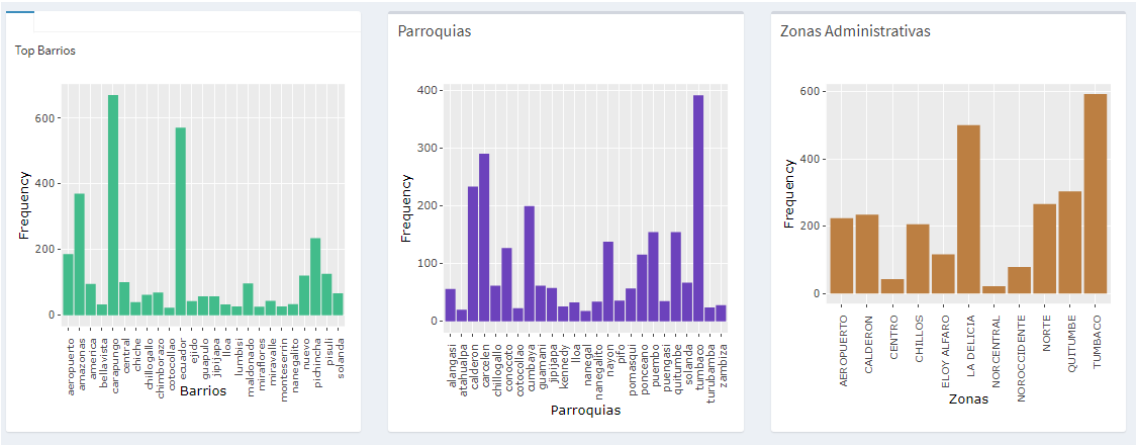

Figura 9. Parte inferior Problemáticas - Generalidades

Por otro lado, en la siguiente sub opción *Map* se puede identificar de manera geográfica el conteo de problemas, como se muestra en la Figura 10, en la parte superior se puede seleccionar si se desea observar el conteo a nivel de barrios,

parroquias o zonas, además permite seleccionar una opción adicional que son las paradas del sistema integrado. Para cada opción se van a visualizar puntos dentro del mapa y un círculo de color, el cual es más grande y rojo a raíz de que existan un mayor número de coincidencias.

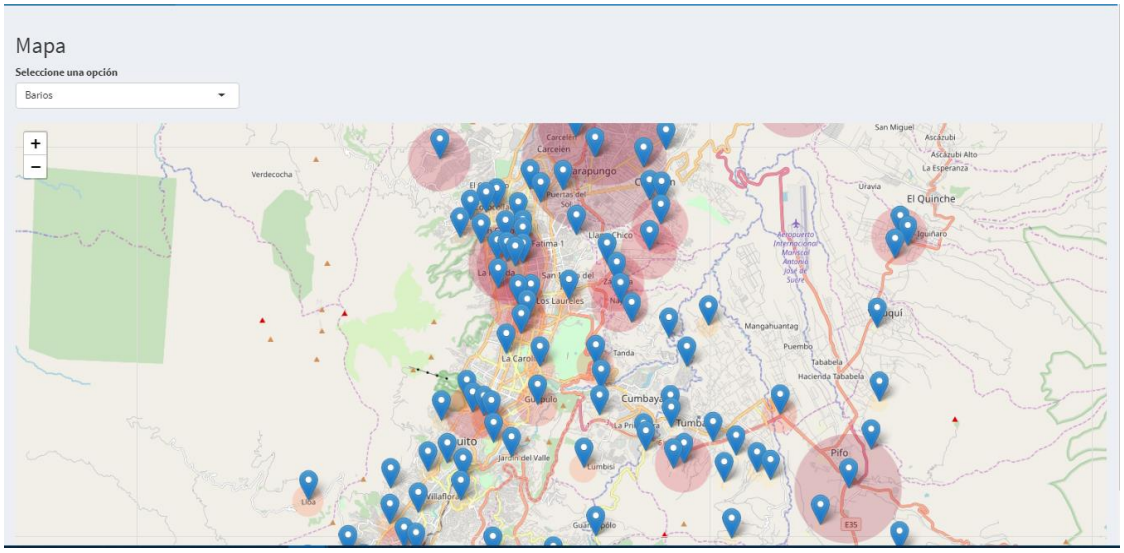

Figura 10. Sub opción de Problemáticas - *Map*

Finalmente, en la última sub opción de Problemática, está Palabras, que muestra el análisis de n-gramas. Como se ve en la figura 11 se puede elegir el número de palabras de 5 hasta 50. Además, existe una pestaña que permite visualizar una tabla de relaciones entre las palabras con su peso respectivo, cuando el gráfico ya no es muy comprensible por la cantidad de relaciones.

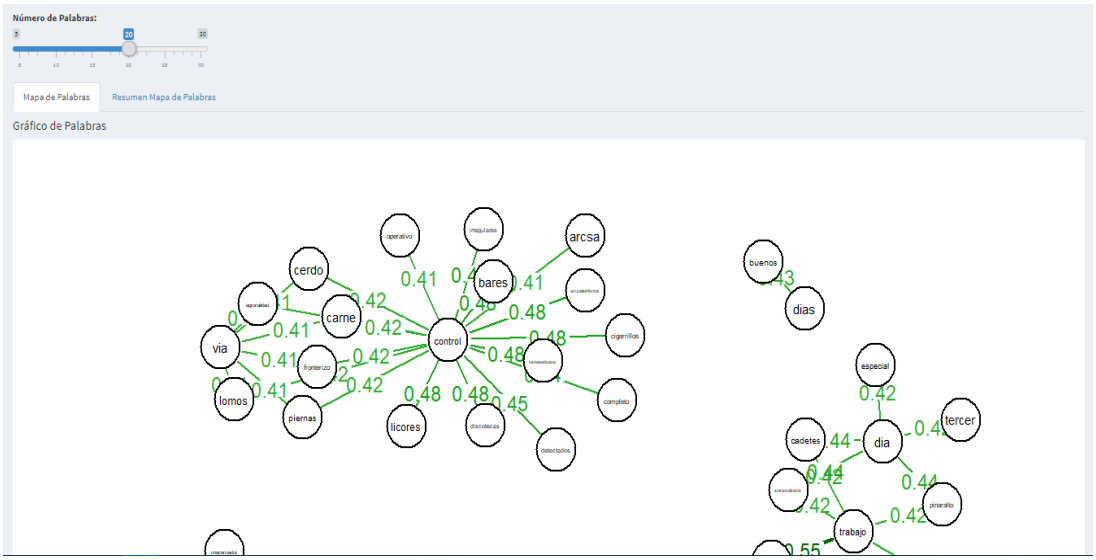

Figura 11. Sub opción de Problemáticas – Palabras

Una vez finalizado el desarrollo de la aplicación, la misma es publicada utilizando los servidores de Shiny, que de manera gratuita permite alojar la aplicación para que esta puede ser accedida desde cualquier lugar y terminal. Una vez publicada, Shiny nos indica cual es el enlace para acceder, en este caso se puede acceder a través del siguiente enlace: https://juanviana.shinyapps.io/SmartCityUIO_App/

4. ANÁLISIS DE RESULTADOS

En esta sección se muestra la información obtenida después de realizar los diferentes análisis.

4.1. Generalidades

Se realizó una obtención de datos generales, el total de tuits y el número de usuarios únicos, estos datos nos permiten conocer el total de datos con lo que se está trabajando para cada uno de los servicios desde que empezó la recolección el 1 de enero hasta el 15 de junio de 2019. En la Figura 5 se observa de izquierda a derecha, de arriba hacia abajo los gráficos de: Todos, Agua, Luz,

Movilidad, Seguridad y Transporte. Se puede observar que uno de los servicios con mayor cantidad de tuits recolectados es Seguridad, con más de 10 mil tuits, seguido por transporte público con 5476 tuis. Por otro lado, el número total de usuarios único es aproximadamente 12 mil usuarios.

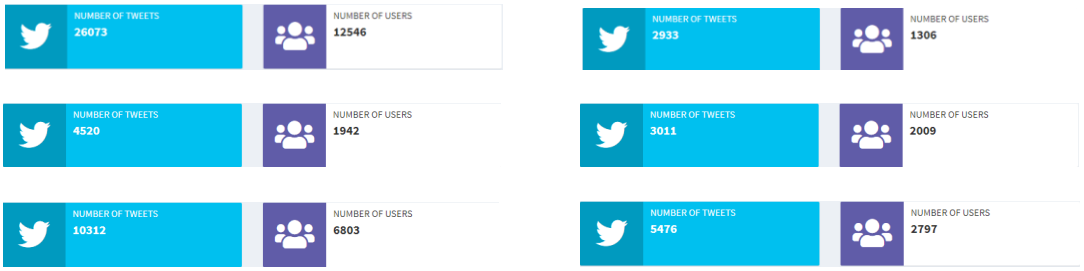

Figura 12. Información Global.

Se observa en la Figura 13, la recopilación diaria de tuits desde enero hasta junio para todos los temas, existen picos en ciertos días de lo cual podemos inferir que existió algo fuera de lo normal, como un accidente grave o un corte de servicio masivo.

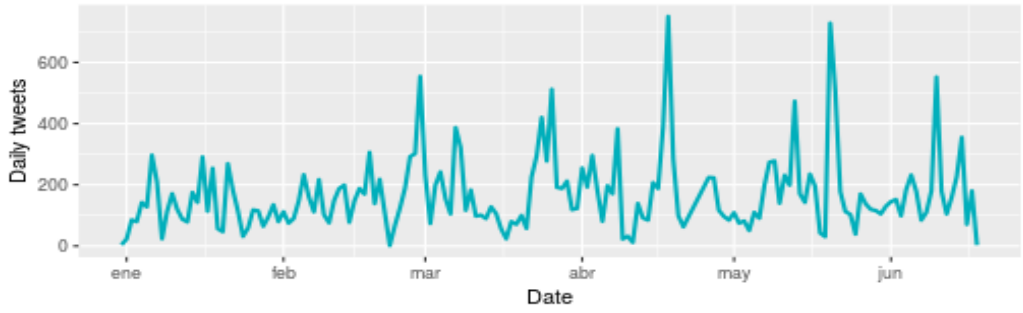

Figura 13. Número de tuits recopilado diariamente durante un periodo de tiempo.

En el resumen mensual que muestra en la Figura 14 se puede observar de izquierda a derecha, de arriba hacia abajo los gráficos de: Agua, Luz, Movilidad, Seguridad y Transporte. Aquí se puede ver que el número de tuits puede variar para cada servicio, esto se puede relacionar con eventos atípicos o el número de días laborales, etc. El último mes es el menor debido a que solo se recopilo hasta la mitad de mes.

Figura 14. Número de tuits mensuales.

Si se observa la Figura 15, se puede ver el número de tuits de manera semanal, aquí hay como apreciar que existen semanas con mayor número de tuits y que tiende a variar en el trascurso del tiempo

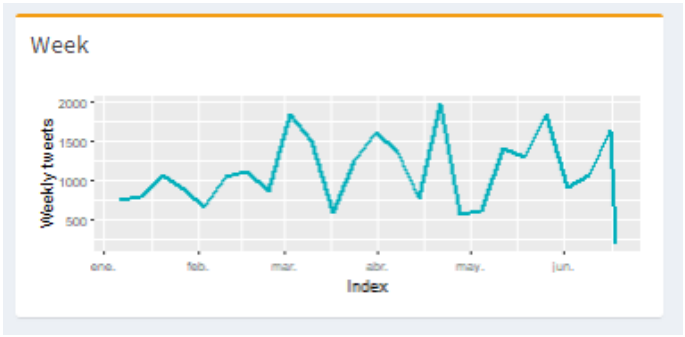

Figura 15. Número de tuits semanales.

En la Figura 16 se ve el día de la semana con mayor número de tuits, en este caso se puede apreciar que el día con mayor número tuits es el día lunes con más de 4 500 tuits, el resto de los días de la semana tiene datos muy similares, aproximadamente 4 mil tuits, excepto sábados y domingos donde este número

se reduce.

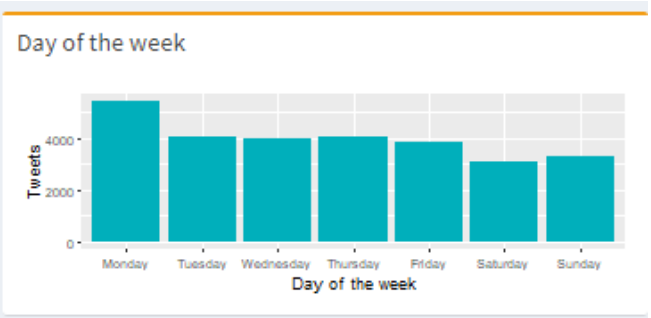

Figura 16. Número de tuis por día de la semana.

Con respecto al número de hashtags y menciones se puede identificar la tendencia que existe con respecto al uso de estas opciones dentro de Twitter. Se aprecia que es más frecuente colocar hashtags a menciones en los tuits, pero por otro lado un tuit suele contener un mayor número de menciones que hashtags. En la Figura 17 se observa la frecuencia de menciones y hashtags por tuits, en el caso de las menciones tenemos que casi 15 000 tuits tienen por lo menos una mención por tuit y aproximadamente 7 500 tuits tiene por lo menos dos menciones por tuits, por el otro lado en el caso de los hashtags, aproximadamente 25 000 tuits tienen por lo menos una etiqueta sin embargo el número de tuits que utilizan dos hashtags es menor a 2 500.

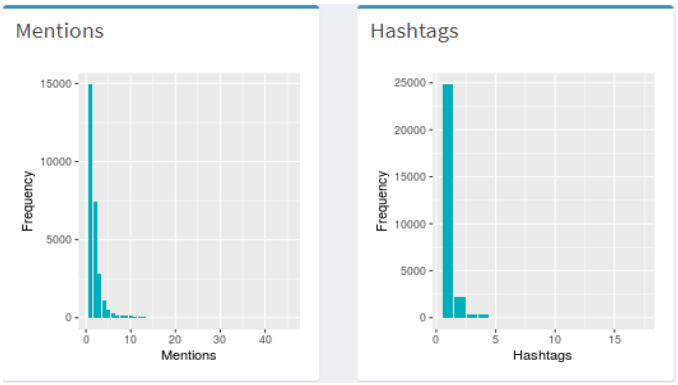

Figura 17. Número de hashtags y menciones por tuit

Además, se puede apreciar en la Figura 18 que los dispositivos con mayor popularidad para realizar un tuit son celulares con sistema operativo Android o

iOS, siendo Android significativamente mayor. Después de estos dos se encuentra a la aplicación web, comúnmente utilizada en computadoras. Esta tendencia se repite para cada uno de los servicios en cuestión.

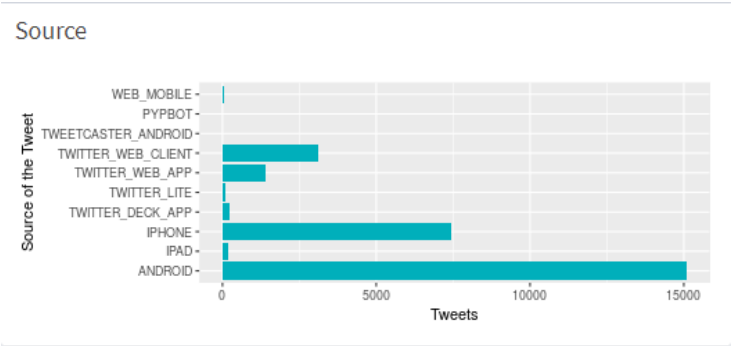

Figura 18. Dispositivos más utilizados para generar los tuits

4.2. Problemáticas Generales

En este punto se puede observar el número de palabras más repetidas para cada uno de los servicios, lo que se busca encontrar tanto problemáticas como ubicaciones conflictivas.

En la Figura 19 se puede identificar que palabras son las más utilizadas en lo tuits recopilados. Dentro de las primeras palabras se encuentra: tráfico, buses, bus. Permitiendo inferir que en más de uno de los servicios se está hablando de estos temas.

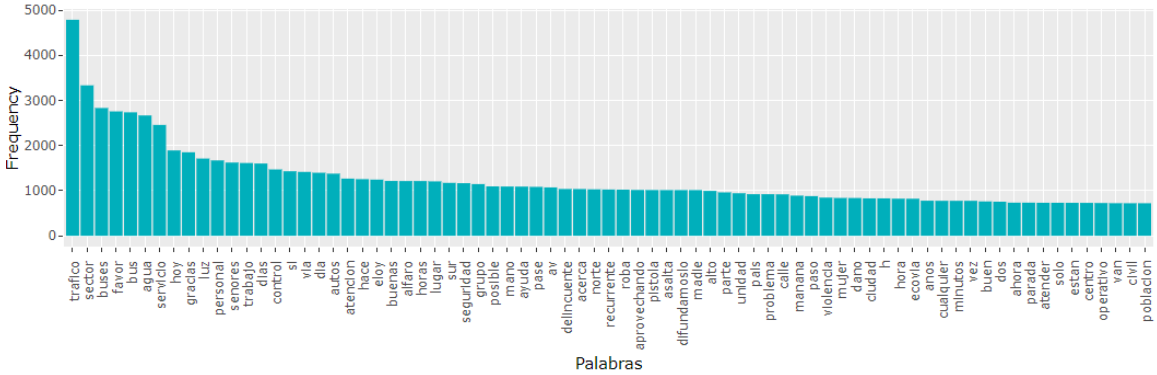

Figura 19. Conteo de Palabras

Después de la tokenización general, es necesario realizar un análisis más profundo para poder comparar por cada uno de los servicios la frecuencia que existe en cada uno de los barrios, parroquias y zonas de la ciudad.

4.2.1. Agua

En la Figura 20 se puede visualizar las frecuencias a nivel de barrios, Carapungo es mencionado más de 500 veces y difiere mucho de los otros. Lo que permite inferir que este barrio puede tener problemas serios con respecto a este servicio.

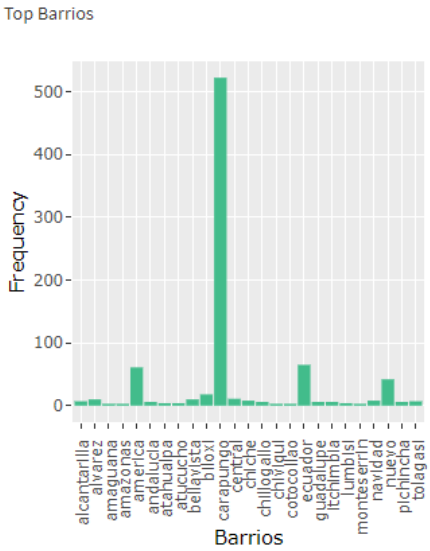

Figura 20. Numero de coincidencias por barrios en el servicio de agua potable

Al observar en la Figura 21, el número de coincidencias a nivel de parroquias se muestra que Calderon, Tumbaco y Puembo tienen la frecuencia más alta, estos son barrios que se encuentran en los alrededores del casco urbano. Luego las parroquias son agrupadas por zonas, aquí lo que se observa es que Tumbaco, que agrupa las parroquias de Cumbayá y Tumbaco es la más mencionada, junto a Aeropuerto y Calderón.

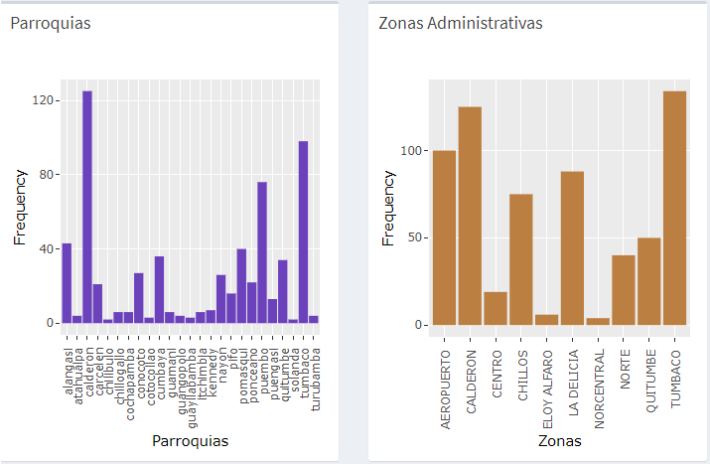

Figura 21. Numero de coincidencias por parroquias y zonas en el servicio de agua potable

4.2.2. Energía Eléctrica

En la frecuencia de barrios, la Figura 22 muestra que la palabra Pichincha es la más mencionada, sin embargo, no es una palabra muy clara, ya que esta misma puede referirse a la provincia y no al barrio como tal. Pero aparte de esto, también barrios como Carapungo y Monteserrín son mencionados de manera significativa.

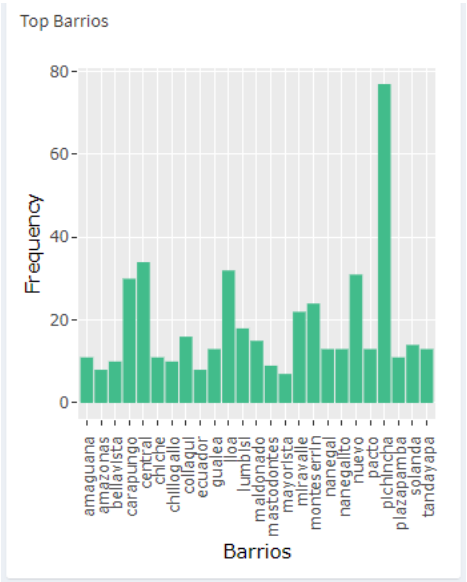

Figura 22. Numero de coincidencias por barrios en el servicio de energía eléctrica

La Figura 23 permite visualizar el número de coincidencias a nivel de parroquias, donde claramente se evidencia que Tumbaco es mencionado más de 200 veces a lo largo de estos 6 meses de recopilación de datos. También, dentro de los más mencionas están: Ponceano, Puembo, Conocoto, Cumbayá y Calderon. De igual manera la Figura 23 permite identificar el resumen por zona donde se comprueba el análisis que se realizó con las parroquias, donde Tumbaco se muestra como una zona crítica para el servicio de energía eléctrica.

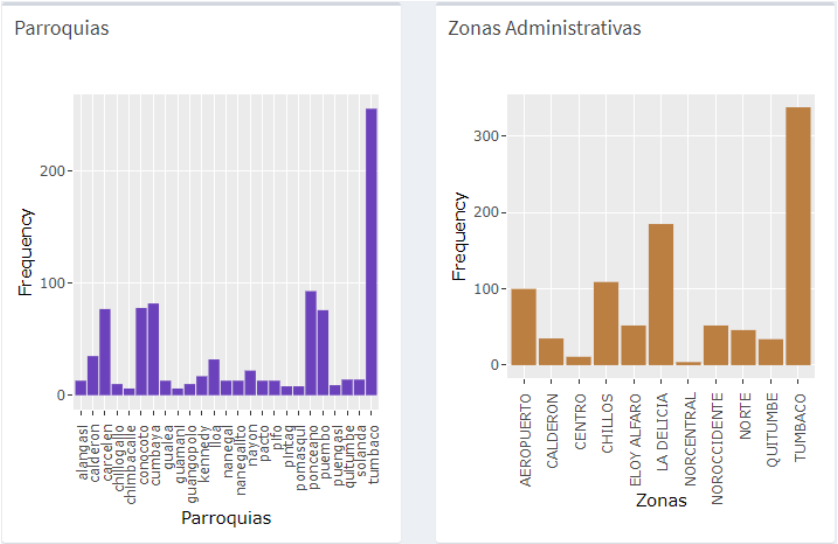

Figura 23. Numero de coincidencias por parroquias y zonas en el servicio de energía eléctrica

4.2.3. Movilidad

Para el tema de movilidad se va a analizar las frecuencias en parroquias y zonas. Esto debido a que los nombres de los barrios son muy similares a los nombres de las calles.

Podemos identificar en la Figura 24 que parroquias como Calderón, Calacalí y Nayón muestran inconvenientes con la movilidad. El resultado a nivel de zonas, donde se ve que existe una frecuencia similar entre Calderón, La Delicia, Norte y Tumbaco. Sin embargo, la Figura 24 no nos permite identificar claramente donde podrían existir inconvenientes con la movilidad, por eso se realiza una

comparación de frecuencias a nivel de ejes viales.

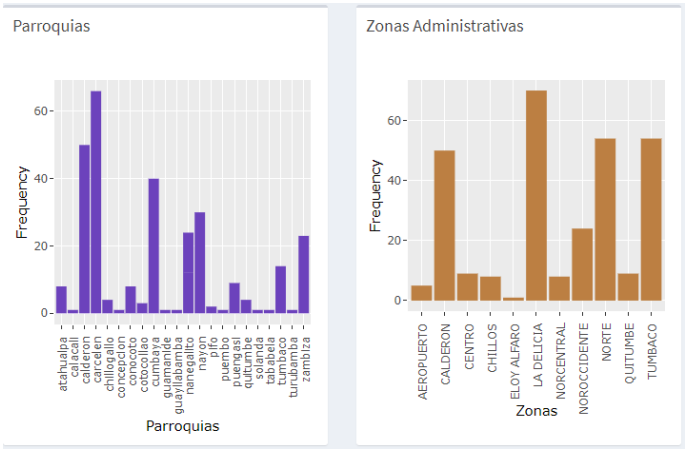

Figura 24. Numero de coincidencias por parroquias y zonas en el servicio de movilidad

Al realizar la comparación a nivel de los ejes viales, las palabras con mayor número de coincidencias como se observa en la Figura 25, son: Simón, Bolívar, Sucre. Muy por debajo de estas están palabras como: Shyris, Amazonas, Carcelén, Carapungo, entre otras. Esto nos permite identificar los ejes viales de la ciudad con mayor número de inconvenientes son la Av. Simón Bolívar y la Mariscal Sucre. Además, resaltan calles como Amazonas y Shyris, la cuales atraviesan el hipercentro de la ciudad.

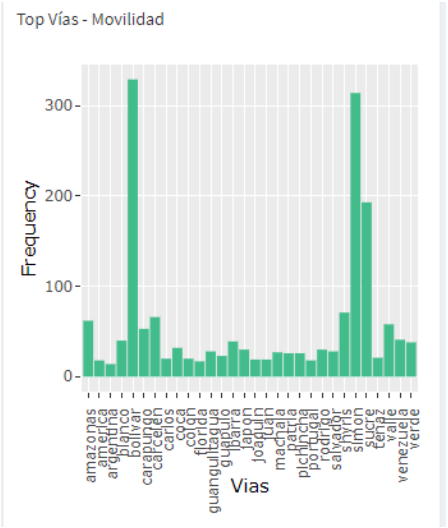

Figura 25. Numero de coincidencias por ejes viales en el servicio de movilidad.

4.2.4. Seguridad

Para le seguridad de igual manera se va a analizar las parroquias y las zonas de la ciudad. En la Figura 26 se observa que parroquias como Jipijapa, Cumbayá y Quitumbe tienen la frecuencia más alta y en la agrupación por zonas se refleja lo mismo. Sin embargo, estas palabras no fueron mencionados más de 50 veces y dentro de un total de más de 10 mil tuits por ende se convierte en algo poco significativo.

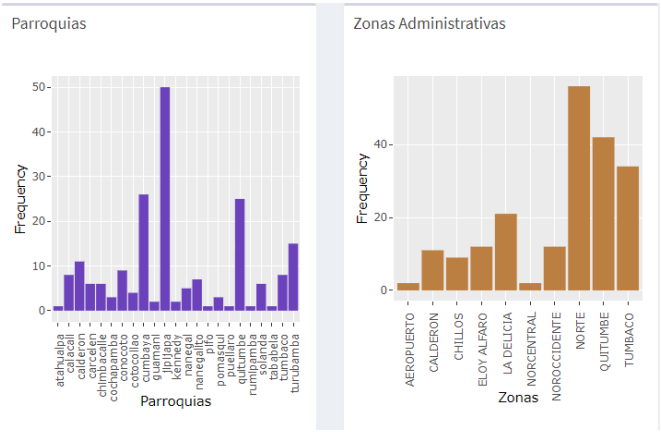

Figura 26. Numero de coincidencias por parroquias y zonas en el servicio de seguridad

4.2.5. Transporte Público

Para empezar, a nivel de parroquias, la Figura 27 nos permite identificar que la más mencionada es Carcelén, seguida de Quitumbe, Nayón y Guamaní. Esto ya permite identificar que estas serían las zonas que experimentan mayor número de irregularidades con respecto al servicio, de igual manera, se puede observar por zonas, donde se confirma lo mencionado

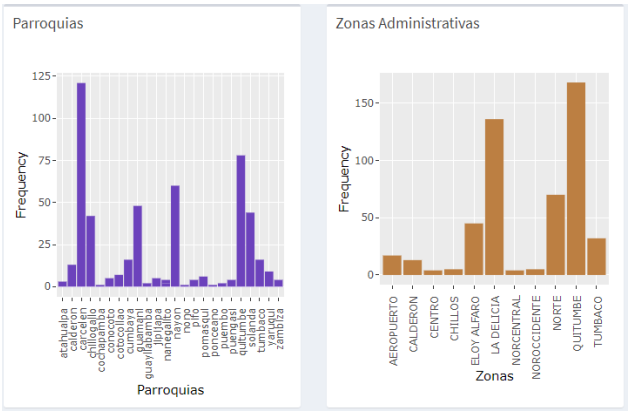

Figura 27. Numero de coincidencias por parroquias y zonas en el servicio de transporte público

Además de estas comparaciones se realizaron gráficos para determinar el número de veces que una parada de bus es mencionada. Para esto el proceso es divido en dos. Primero las paradas de buses y segundo las paradas del sistema integrado de transporte.

Uno de los problemas que se identifica al observar los resultados de la Figura 28 es que en las paradas de buses se puede llegar a conocer en que calles de Quito podrían estar los inconvenientes, pero puede que este no sea tan exacto. Esto debido a que, en la base municipal de paradas, estas están identificadas por la calle principal más no por un nombre, lo que hace que su ubicación sea más complicada.

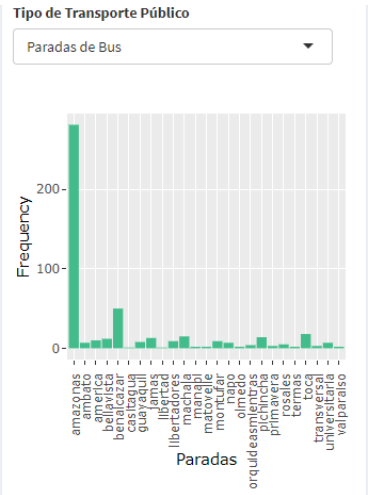

Figura 28. Numero de coincidencias por paradas de bus en el servicio de transporte público

Por otro lado, la Figura 29 nos muestra que en las siguientes paradas del sistema integrado existe una alta frecuencia: Bellavista, Solanda y Chillogallo; pudiendo ser estas las que presentan un mayor número de inconvenientes.

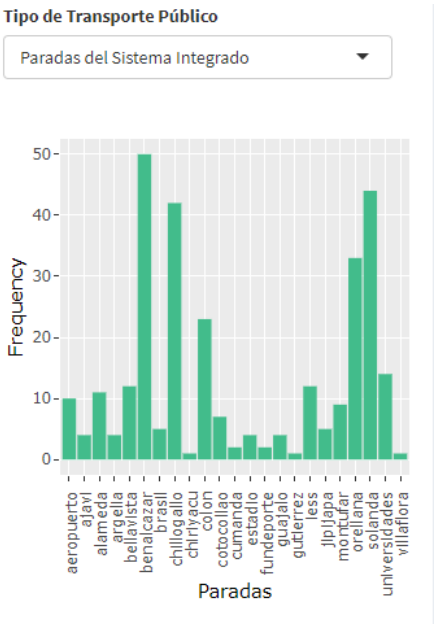

Figura 29. Numero de coincidencias por paradas del sistema integrado en el servicio de transporte público

4.3. Mapa de Problemáticas

Después de haber establecido la frecuencia por barrios, parroquias y zonas, el ubicar estos geográficamente nos permite reconocer donde estarían los focos con mayor número de problemas.

En el análisis barrial, encontramos una dificultad debido a que existen varios barrios con los mismos nombres a pesar de haber encontrado coincidencias es difícil distinguir a cuál de los barrios se hace referencia. Por ejemplo, esto sucede con barrios como: Bellavista, Central, etc. Existen varios barrios a lo largo de la ciudad con este nombre y están geográficamente separados.

En las parroquias podemos visualizar que para cada servicio tenemos diferentes puntos de calor, sin embargo, se puede notar una tendencia en estos focos y es que en el caso de servicios básicos se encuentran en los alrededores al caso urbano o apartados de los hipercentros de la ciudad, como se observa en la Figura 30 de izquierda a derecha de, de arriba hacia abajo los gráficos de: Todos, Agua, Luz, Movilidad, Seguridad y Transporte.

Estos mapas son muy buenos para temas puntuales, como agua o luz debido a que en los tuits se encuentra el nombre de un barrio, parroquia o zona. Sin embargo, en el tema de movilidad muchas veces se menciona una calle restringiendo apreciar correctamente donde se encuentra la mayor cantidad de problemas.

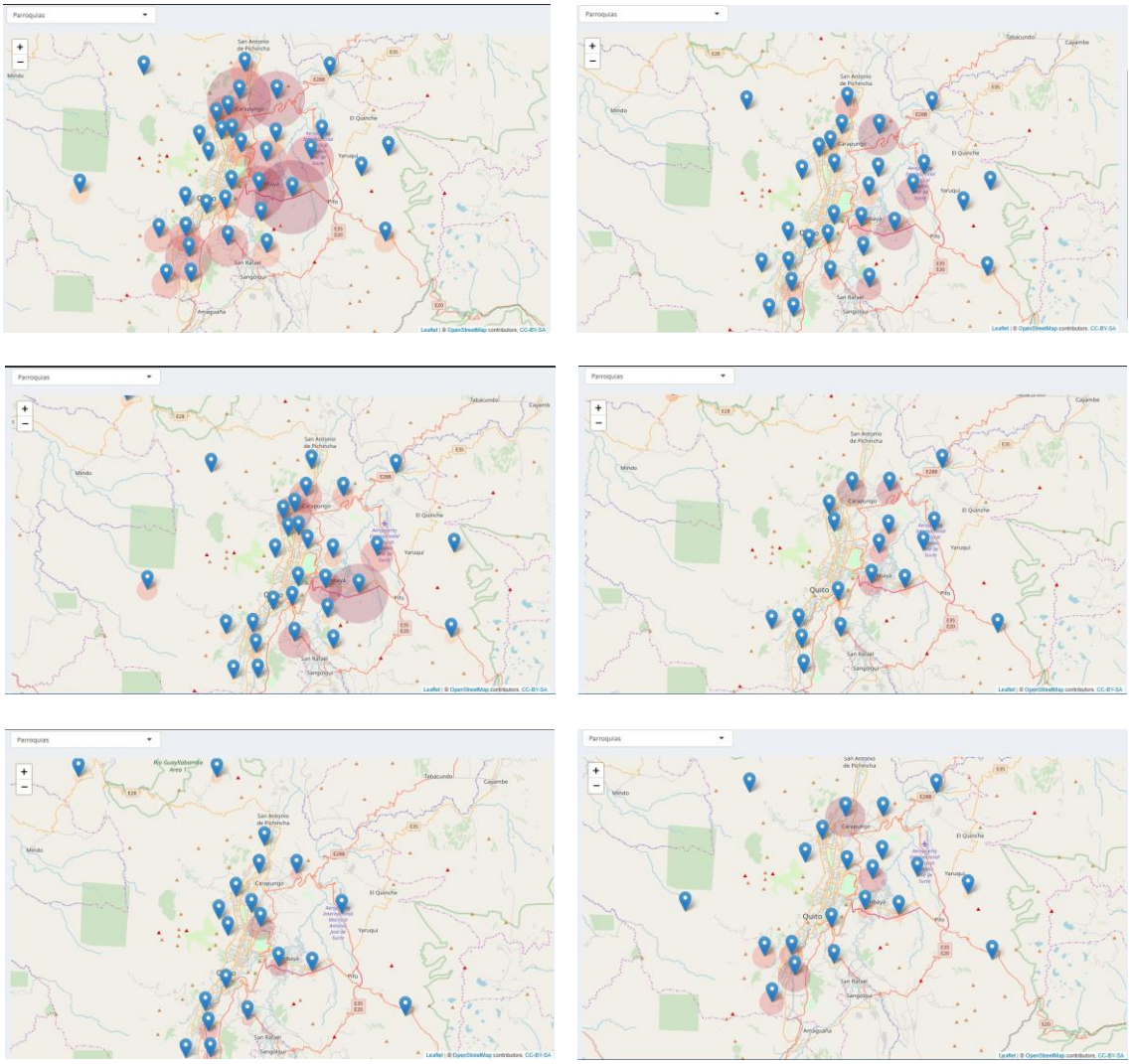

Figura 30. Mapa de coincidencias de parroquias.

Lo que respecta al transporte, en la Figura 31 se puede visualizar las paradas del sistema integrado de transporte, no siempre las paradas son mencionadas en los tuits lo que reduce la exactitud, pero dejando esto de lado, si se puede ver que las paradas más transitadas están cerca de puntos de interés, como universidades, centros comerciales o complejos de oficinas.

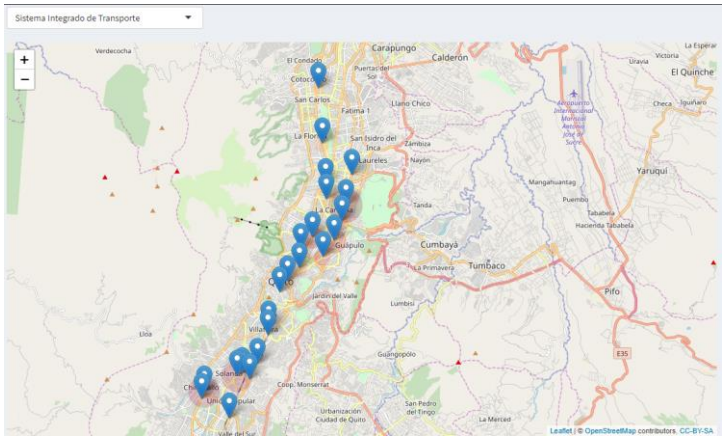

Figura 31. Mapa de coincidencias de paradas del Sistema Integrado de Transporte.

4.4. Red de Palabras

La red de palabras nos permite visualizar como a lo largo del texto de un tuit las palabras se relacionan entre sí. Esto permite identificar problemáticas o los temas más mencionados. La Figura 32 muestra un ejemplo de los gráficos realizados en el aplicativo, el objetivo de realizar este análisis es que se puede observar la relación que existe entre las palabras mencionadas dentro de un tuit, es decir permite identificar de qué manera los usuarios escriben el tuit y así poder visualizar la relación entre palabras y el peso que tienen con respecto a otras relaciones, como ejemplo en la Figura 32 se observa la relación “control – licores” con una ponderación 0.48

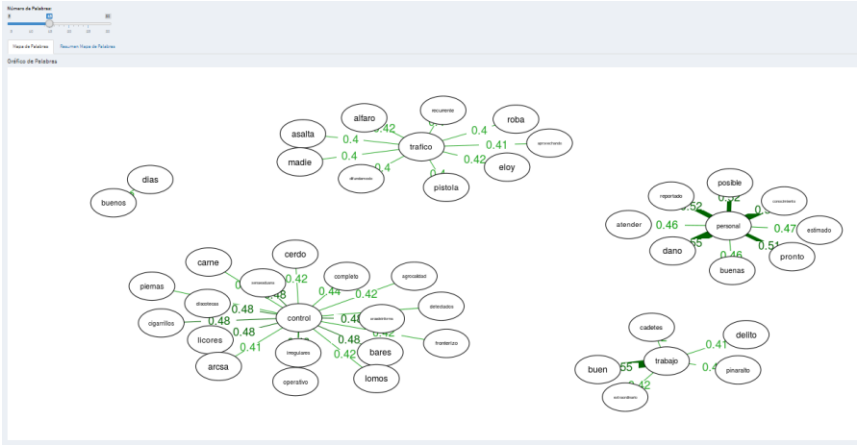

Figura 32. Relación entre palabras

4.4.1. Agua

En la Figura 33 se observa algunos sectores como: Pampa, Armenia, Liga. Aquí ya se empiezan a encontrar inconvenientes en el servicio como “servicio pésimo”, “carapungo sinagua”, “servicio suspensión” Esto nos brinda una idea de cuáles son las opiniones y problemas son las que enfrenta este servicio.

Resumen Relación de Palabras

from	to	weight
favor	ayuda	0.40
servicio	potable	0.30
servicio	suspension	0.30
servicio	informa	0.28
servicio	normalizara	0.28
favor	colaboren	0.27
servicio	transcurso	0.26
servicio	debido	0.23
sector	pampa	0.23
servicio	informacion	0.22
favor	armenia	0.22
servicio	transcurso	0.26
servicio	debido	0.23
sector	pampa	0.23
servicio	informacion	0.22
favor	armenia	0.22
agua	potable	0.21
servicio	barrios	0.21
agua	senores	0.20
servicio	pesimo	0.20
carapungo	sinagua	0.20
sector	ayuda	0.20
sector	alexismoncayo	0.20
sector	liga	0.20

Figura 33. Relación entra palabras del servicio de agua potable

4.4.2. Energía Eléctrica

Analizando las relaciones que se observan la Figura 34 se puede inferir varios temas. Primero los inconvenientes en el servicio se presentan con mayor probabilidad en la tarde y en la noche. Además, se puede apreciar que siempre que existe una queja, hay una respuesta de por medio, dado que se encuentra palabras como: “personal-notificado”, “dano-reportado”; con la misma ponderación a la relación que muestra el problema. Lo que nos indica que a pesar de que el servicio muestra inconvenientes el mismo responde a las quejas de los usuarios.

Resumen Relación de Palabras

from	to	weight
posible	pronto	0.78
dano	reportado	0.77
posible	reportado	0.76
atender	reportado	0.70
dano	pronto	0.69
personal	conocimiento	0.66
buenas	tardes	0.66
buenas	noches	0.65
atender	conocimiento	0.65
personal	reportado	0.64
posible	estimado	0.64
personal	pronto	0.63
buenas	reportado	0.62
personal	estimado	0.60
posible	conocimiento	0.60
buenas	estimado	0.58
dano	estimado	0.58
atender	pronto	0.58
atender	estimado	0.55
buenas	pronto	0.53
dano	conocimiento	0.51
buenas	conocimiento	0.49
posible	sitio	0.49
favor	ayuda	0.47
posible	noches	0.46
dano	noches	0.46
dano	sitio	0.46
personal	sitio	0.45
posible	tardes	0.44
posible	estimada	0.44
posible	notificado	0.43
buenas	sitio	0.42
personal	noches	0.41
personal	estimada	0.40
buenas	estimada	0.40
dano	estimada	0.40
atender	noches	0.40
personal	tardes	0.39
atender	tardes	0.38
personal	notificado	0.37

Figura 34. Relación entra palabras del servicio de energía eléctrica

4.4.3. Movilidad

Dentro de las palabras que más llaman la atención en la Figura 35 esta simón y bolívar, esto debido a la Avenida Simón Bolívar y su importancia e influencia que tiene en la movilidad de la ciudad. Además, la Figura 35 nos permite identificar tramos o el motivo del inconveniente. Por ejemplo, tramos cerca de la Panamericana, Carcelén y Calderón son los más mencionados. También, vemos que muchas veces se relaciona con la palabra restricción, lo que permite inferir que se refiere a la circulación y un factor interesante es que con un peso de 0.5 está la palabra accidente, lo que nos muestra que gran parte de las veces la avenida es mencionada por ese motivo.

Resumen Relación de Palabras

from	to	weight
simon	blanco	0.80
simon	restriccion	0.80
bolivar	blanco	0.79
bolivar	restriccion	0.79
simon	registrado	0.75
norte	sur	0.75
bolivar	registrado	0.74
simon	pueblo	0.74
bolivar	pueblo	0.73
simon	calderon	0.70
simon	panamericana	0.70
bolivar	calderon	0.69
bolivar	panamericana	0.69
simon	bajada	0.65
bolivar	bajada	0.64
simon	carcelen	0.60
simon	reporta	0.60
bolivar	carcelen	0.59
bolivar	reporta	0.59
norte	sentido	0.58
norte	llantas	0.55
norte	quema	0.55
simon	accidente	0.54
bolivar	accidente	0.53
norte	legarda	0.53
norte	lento	0.51
simon	hacia	0.50
bolivar	hacia	0.49
simon	lento	0.49
bolivar	lento	0.48
via	llantas	0.47
via	quema	0.47
via	legarda	0.46
norte	atencion	0.46
sector	lento	0.45
bolivar	realizan	0.44
simon	realizan	0.44
via	atencion	0.43
simon	cierres	0.42
norte	sucre	0.42

Figura 35. Relación entra palabras del servicio de movilidad

4.4.4. Seguridad

Al observar la Figura 36 se puede ver que la relación de palabras obtenidas se muestra un tema que probablemente se ha vuelto frecuente en la ciudad que son asaltos en grupos, que estos están tomando lugar en situaciones de tráfico, donde existen asaltos a carros con pistolas y puede que esto ese sucediendo sobre la Avenida Eloy Alfaro y es recurrente.

Resumen Relación de Palabras

from	to	weight
grupo	acerca	0.94
grupo	alfaro	0.94
grupo	aprovechando	0.94
grupo	asalta	0.94
grupo	delincuente	0.94
grupo	difundamoslo	0.94
grupo	eloy	0.94
grupo	madie	0.94
grupo	pistola	0.94
grupo	recurrente	0.94
grupo	roba	0.94
grupo	mano	0.93
grupo	pase	0.93
autos	aprovechando	0.85
autos	asalta	0.85
autos	difundamoslo	0.85
autos	madie	0.85
autos	pistola	0.85
autos	recurrente	0.85
autos	recurrente	0.85
autos	acerca	0.84
autos	alfaro	0.84
autos	delincuente	0.84
autos	eloy	0.84
autos	roba	0.84
autos	mano	0.83
autos	pase	0.83
trafico	aprovechando	0.82
trafico	asalta	0.82
trafico	difundamoslo	0.82
trafico	madie	0.82
trafico	pistola	0.82
trafico	recurrente	0.82
trafico	acerca	0.81
trafico	alfaro	0.81
trafico	delincuente	0.81
trafico	eloy	0.81
trafico	roba	0.81
trafico	mano	0.80

Figura 36. Relación entra palabras del servicio de seguridad

4.4.5. Transporte

En temas de transporte, como se observa en la Figura 37, las relaciones encontradas son sobre temas de irresponsabilidad en el servicio, esto se puede inferir con palabras como: choque, carreras y rebasa. Aquí podemos observar que gran parte del problema en el servicio es con respecto a la calidad prestada por las unidades de transporte público.

Resumen Relación de Palabras

from	to	weight
casi	choca	0.90
casi	dimos	0.89
casi	fresco	0.89
casi	suelo	0.89
casi	vira	0.89
casi	paquisha	0.88
casi	rebasa	0.88
casi	puntos	0.86
casi	voy	0.85
casi	pasaje	0.83
casi	quieren	0.82
casi	evitar	0.81
casi	altura	0.80
casi	choque	0.79
casi	carreras	0.72
casi	amazonas	0.71
casi	ahora	0.71
casi	mismo	0.70
hace	subir	0.69
casi	subir	0.66
casi	sentido	0.65
hace	dimos	0.63
hace	suelo	0.63
hace	vira	0.63
hace	paquisha	0.62
hace	fresco	0.62
hace	rebasa	0.62
hace	puntos	0.60
casi	van	0.59
hace	voy	0.59
hace	pasaje	0.59
hace	quieren	0.59
unidad	dimos	0.58
unidad	fresco	0.58
unidad	rebasa	0.58
unidad	suelo	0.58
unidad	vira	0.58
unidad	paquisha	0.57
unidad	puntos	0.56

Figura 37. Relación entra palabras del servicio de transporte público.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Con el trabajo realizado se puede concluir que cada servicio tiene sectores con un mayor número de conflictos, que en la mayoría de los casos están situados alrededor de los hipercentros del distrito. Este tipo de información permite identificar donde están las zonas con mayor conflicto y el número de inconvenientes que podría respaldar la toma de decisiones.

La captura de este tipo de información puede ayudar en la mejora del servicio prestado por la entidad pública, dado que este medio se torna en un canal de denuncia y reclamo por el cual se puede gestionar de una manera más ágil donde se encuentran los inconvenientes. Hablando también de manera global ya que se puede identificar si el problema está a nivel barrial, parroquial o incluso a un nivel zonal.

Finalmente lograr que el Distrito Metropolitano de Quito sea una ciudad inteligente a través del uso de las redes sociales como Twitter para detección de problemas puede ser de mucha ayuda, más si se proyecta a la detección en tiempo real de la información. Se pueden utilizar estos canales para detectar inconvenientes dentro de la ciudad y reducir tiempos de respuesta. Este tipo de análisis nos da una visión de problemas o lugares que a veces no se toman en cuenta, sin embargo, esto no solo depende de una ciudad con infraestructura, también de una ciudanía abierta al uso de estas tecnologías y que sean utilizadas por las entidades Municipales.

5.2. Recomendaciones

Es importante tomar en cuenta que la información capturada debe ser manejada de la mejor manera, no solo por los términos y condiciones que impone Twitter

para poder utilizar el servicio, también por que se maneja información que a pesar de ser pública puede ser sensible, por lo que en estos casos lo mejor es anonimizar los datos.

Dentro de todo este proceso existieron algunas dificultades como son las restricciones que Twitter presenta para utilizar la API, es aquí donde vale recalcar que un proyecto que logre recopilar datos en tiempo real puede aumentar la eficiencia de la ilustración de los datos y tener una mejor reacción antes las diferentes eventualidades. También, vale la pena recalcar que en el distrito metropolitano hay barrios con nombres idénticos o similares en distintas ubicaciones geográficas lo que impide ubicar correctamente a este nivel. Además, el usuario debe ser lo más preciso al momento de escribir el lugar donde tiene el problema ya que es a partir del texto que se puede detectar inconveniente.

REFERENCIAS

Almeida, V., Moreira, E., & Danilo, D. (2018). *Humane Smart Cities : The Need for Governance. IEEE Internet Computing,* (April), 91–95. https://doi.org/10.1109/MIC.2018.022021671

Beysolow II, T. (2018). *Applied Natural Language Processing with Python*. San Francisco: *APRESS*. https://doi.org/10.1007/978-1-4842-3733-5

Chambers, J., & Hand, D. (2008). *Software fo Data Analysis*. (J. Chambers, D. Hand, & W. Härdle, Eds.). New York: *Springer International Publishing*. Recuperado de https://link-springer-com.bibliotecavirtual.udla.edu.ec/content/pdf/10.1007%2F978-0-387-75936-4.pdf

Distrito Metropolitano de Quito. (2015). Paradas Transporte Público. Recuperado el 26 de mayo de 2019, de http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/descargashp/shp/Descargables/Estaciones_BRT.zip

Distrito Metropolitano de Quito. (2016). Estaciones BRT Sistema Integrado. Recuperado el 26 de mayo de 2019, de http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/descargashp/shp/Descargables/Estaciones_BRT.zip

Distrito Metropolitano de Quito. (2018). Ejes vias. Recuperado el 26 de mayo de 2019, de http://gobiernoabierto.quito.gob.ec/wp-content/uploads/documentos/descargashp/shp/Descargables/Vias.zip

E. Banchs, R. (2013). *Text Mining with MATLAB*. New York: *Springer International Publishing*. Recuperado de https://link-springer-com.bibliotecavirtual.udla.edu.ec/content/pdf/10.1007%2F978-1-4614-4151-9.pdf

Garimella, K., Mathioudakis, M., Gionis, A., & Morales, G. D. F. (2016). *Quantifying Controversy in Social Media*, 33–42. https://doi.org/10.1145/1235

Garimella, K., Weber, I., & Tech, G. (2016). *Quote RTs on Twitter : Usage of the New Feature for Political Discourse*, 200–204. Recuperado el 21 de julio, de 2019 de https://ingmarweber.de/wp-content/uploads/2016/05/Quote-RTs-on-Twitter-Usage-of-the-New-Feature-for-Political-Discourse.pdf

Harmon, R. R., & Lee, M. R. (2016). *IT in Smart Cities*, 14–17. Recuperado el 22 de

julio de 2019, de https://publications.iadb.org/publications/english/document/International-Case-Studies-of-Smart-Cities-Songdo-Republic-of-Korea.pdf

Kapadia, S. (2019). *Introduction to Language Models: N-Gram*. Recuperado el 25 de julio de 2019, de https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9

Kargın, Y., Ivanova, M., Zhang, Y., Manegold, S., & Kersten, M. (2013). *Lazy ETL in Action : ETL Technology Dates Scientific Data*, *6*(12), 1286–1289. Recuperado de https://dl-acm-org.bibliotecavirtual.udla.edu.ec/citation.cfm?id=2536297

Ministerio de Telecomunicaciones y de la Sociedad de la Información. (2015). 91% de ecuatorianos utiliza las redes sociales en su teléfono inteligente. Recuperado de https://www.telecomunicaciones.gob.ec/91-de-ecuatorianos-utiliza-las-redes-sociales-en-su-telefono-inteligente/

OPENMINTED COMMUNICATIONS. (2018). *TEXT MINING 101.* Recuperado el 27 de julio de 2019, de http://openminted.eu/text-mining-101/

R Foundation. (2019). *The R Project for Statistical Computing*. Recuperado el 15 de junio de 2019, de https://www.r-project.org/

Secretaría de Territorio Hábitat y Vivienda. (2017). Barrio Sector. Recuperado el 26 de mayo de 2019, de http://geo.quito.gob.ec:8080/geonetwork/srv/spa/catalog.search#/metadata/939c016c-56c1-4505-80f4-f258862a132f

Suma, S., Mehmood, R., & Albeshri, A. (2018). *Smart Societies, Infrastructure, Technologies and Applications* (Vol. 224). *Springer International Publishing*. https://doi.org/10.1007/978-3-319-94180-6

Twitter. (2017). *Tweet Updates*. Recuperado el 16 de junio de 2019, de https://developer.twitter.com/en/docs/tweets/tweet-updates.html

Twitter. (2018). Centro de Ayuda, Cómo retwittear. Recuperado el 23 de noviembre de 2018, de https://help.twitter.com/es/using-twitter/how-to-retweet

Twitter. (2019). *Search Tweets*. Recuperado el 16 de junio de 2019, de https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators

ANEXOS

ANEXO 1

FORMATO JSON


```
{
        "contributors": null,
        "coordinates": null,
        "created_at": "Thu Jan 17 02:04:26 +0000 2019",
        "display_text_range": [
            33,
            187
        ],
        "entities": {
            "hashtags": [],
            "symbols": [],
            "urls": [],
            "user_mentions": [
                {
                    "id": 253628616,
                    "id_str": "253628616",
                    "indices": [
                        0,
                        11
                    ],
                    "name": "Obras Quito",
                    "screen_name": "ObrasQuito"
                },
                {
                    "id": 227720098,
                    "id_str": "227720098",
                    "indices": [
                        12,
                        22
                    ],
                    "name": "ximena",
                    "screen_name": "ximenacab"
                },
                {
                    "id": 545416010,
                    "id_str": "545416010",
                    "indices": [
                        23,
                        32
                    ],
                    "name": "AMT Quito",
                    "screen_name": "AMTQuito"
                }
            ]
        },
        "favorite_count": 1,
        "favorited": false,
        "full_text": "@ObrasQuito @ximenacab @AMTQuito Que VIVA LA BUROCRACIA\u00a1\u00a1\u00a1
hasta ahora sigo esperando la soluci\u00f3n a un foco da\u00f1ado de un sem\u00e1foro de una
autopista de alto trafico y velocidad en la misma.",
        "geo": null,
        "id": 1085719465814298624,
        "id_str": "1085719465814298624",
        "in_reply_to_screen_name": "ObrasQuito",
        "in_reply_to_status_id": 1085696977713745920,
        "in_reply_to_status_id_str": "1085696977713745920",
        "in_reply_to_user_id": 253628616,
        "in_reply_to_user_id_str": "253628616",
        "is_quote_status": false,
        "lang": "es",
        "metadata": {
            "iso_language_code": "es",
            "result_type": "recent"
        },
        "place": null,
        "retweet_count": 0,
        "retweeted": false,
        "source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for
```

```
Android</a>",
        "truncated": false,
        "user": {
            "contributors_enabled": false,
            "created_at": "Tue Jan 04 21:39:46 +0000 2011",
            "default_profile": true,
            "default_profile_image": false,
            "description": "Soy Quien Quiero Ser y Lo Ser\u00e9 As\u00ed Por Siempre",
            "entities": {
                "description": {
                    "urls": []
                }
            },
            "favourites_count": 3323,
            "follow_request_sent": false,
            "followers_count": 156,
            "following": false,
            "friends_count": 1126,
            "geo_enabled": false,
            "has_extended_profile": true,
            "id": 234109555,
            "id_str": "234109555",
            "is_translation_enabled": false,
            "is_translator": false,
            "lang": "es",
            "listed_count": 0,
            "location": "Quito",
            "name": "BETO",
            "notifications": false,
            "profile_background_color": "C0DEED",
            "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
            "profile_background_image_url_https":
"https://abs.twimg.com/images/themes/theme1/bg.png",
            "profile_background_tile": false,
            "profile_banner_url": "https://pbs.twimg.com/profile_banners/234109555/1497576522",
            "profile_image_url":
"http://pbs.twimg.com/profile_images/984082344574443523/jnVqPB8g_normal.jpg",
            "profile_image_url_https":
"https://pbs.twimg.com/profile_images/984082344574443523/jnVqPB8g_normal.jpg",
            "profile_link_color": "1DA1F2",
            "profile_sidebar_border_color": "C0DEED",
            "profile_sidebar_fill_color": "DDEEF6",
            "profile_text_color": "333333",
            "profile_use_background_image": true,
            "protected": false,
            "screen_name": "marioalberto81",
            "statuses_count": 6738,
            "time_zone": null,
            "translator_type": "none",
            "url": null,
            "utc_offset": null,
            "verified": false
        }
    }
```

ANEXO 2

TRANSFORMACIONES DE FECHAS

```
CREATE OR ALTER VIEW FECHAS_TRANSPORTE AS
SELECT DISTINCT
        TWEET_CLASIFIED_ID ,
        FECHA_NUEVA AS FECHA,
        DATENAME(DAY,FECHA_NUEVA) DIA,
        SUBSTRING((CONVERT(VARCHAR,FECHA_NUEVA,110)),1,2) MES,
        DATENAME(YEAR,FECHA_NUEVA) ANIO,
        DATENAME(WEEKDAY,FECHA_NUEVA) NOMBRE_DIA,
        DATENAME(MONTH,FECHA_NUEVA) NOMBRE_MES,
        DATENAME(HOUR,FECHA_NUEVA) HORA
FROM (
        SELECT *,dateadd(HOUR, -5, FECHA_COMPUESTA ) AS FECHA_NUEVA
        FROM (
                SELECT *, CONVERT(datetime, CONCAT(ANIO,'-',CONCAT(MES_ANIO_NUM,'-
',CONCAT(DIA_MES, ' ',HORA)))) FECHA_COMPUESTA
                FROM (
                                SELECT
                                        id_tweet AS TWEET_CLASIFIED_ID,
                                        CASE
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Jan'
                                                        THEN '01'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Feb'
                                                        THEN '02'
                                                WHEN SUBSTRING(created_at_tweet, 5,3)= 'Mar'
                                                        THEN '03'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Apr'
                                                        THEN '04'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'May'
                                                        THEN '05'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Jun'
                                                        THEN '06'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Jul'
                                                        THEN '07'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Agu'
                                                        THEN '08'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Sep'
                                                        THEN '09'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Oct'
                                                        THEN '10'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Nov'
                                                        THEN '11'
                                                WHEN SUBSTRING(created_at_tweet, 5,3) = 'Dic'
                                                        THEN '12'
                                        END AS 'MES_ANIO_NUM',
                                        SUBSTRING(created_at_tweet, 9,2) AS 'DIA_MES',
                                        SUBSTRING(created_at_tweet, 27,4) AS 'ANIO',
                                        SUBSTRING(created_at_tweet, 12,8) AS 'HORA'
                                        FROM dbSmartCitiesUIO.dbo.transporte
                                ) AS FECHASTRANSPORTE
                        ) AS FECHASMODIFY
                ) AS FECHASFINAL
GO
```

ANEXO 3

SCRIPT SQL PARA GENERAR VISTA CON TODAS LAS TRANSFORMACIONES

```
CREATE VIEW CONSULTA_TRANSPORTE_COMPLETA
AS
SELECT DISTINCT
        LTRIM(RTRIM(id_user)) AS ID_USER,
        LTRIM(RTRIM(screen_name_user)) AS SCREEN_NAME_USER,
        LTRIM(RTRIM(name_user)) AS NAME_USER,
        LTRIM(RTRIM(id_tweet)) AS ID_TWEET,
        LTRIM(RTRIM(retweet_count_tweet)) AS RT_COUNT_TWEET,
        LTRIM(RTRIM(favorite_count_tweet)) AS FAV_COUNT_TWEET,
        LTRIM(RTRIM(first_3_letters_tweet)) AS FIRST_3_LETTERS_TWEET,
        LTRIM(RTRIM(is_retweet_tweet)) AS IS_RETWEET_TWEET,
                source_tweet_classified AS SOURCE_TWEET,
        LTRIM(RTRIM(hashtags_in_tweet)) AS HASHTAGAS_TWEET,
        LTRIM(RTRIM(number_hashtags_in_tweet)) AS NUM_HASHTAGS_TWEET,
        LTRIM(RTRIM(mentiones_in_tweet)) AS MENTIONES_TWEET,
        LTRIM(RTRIM(number_mentions_in_tweets)) AS NUM_MENTIONES_TWEET,
        LTRIM(RTRIM(rt_tweet_id)) AS RT_ID_TWEET,
        LTRIM(RTRIM(TWEET_CLASIFIED_TEXT)) AS TEXT_TWEET,
        LTRIM(RTRIM(FECHA)) AS FECHA,
                NOMBRE_DIA,
                NOMBRE_MES,
                HORA,
                DIA,
                MES,
                ANIO
                FROM (
                        SELECT *,
                        CASE
                                WHEN is_retweet_tweet = 1
                                        THEN rt_tweet_text
                                ELSE
                                        text_tweet
                        END AS TWEET_CLASIFIED_TEXT,
                        CASE
                                WHEN source_tweet LIKE '%>Twitter for Android<%'
                                        THEN 'ANDROID'
                                WHEN source_tweet LIKE '%>Twitter for iPhone<%'
                                        THEN 'IPHONE'
                                WHEN source_tweet LIKE '%>Twitter Lite<%'
                                        THEN 'TWITTER_LITE'
                                WHEN source_tweet LIKE '%>Facebook<%'
                                        THEN 'FACEBOOK'
                                WHEN source_tweet LIKE '%>PERSICOPE<%'
                                        THEN 'PERSICOPE_LIVE_VIDEO'
                                WHEN source_tweet LIKE '%>TweetDeck<%'
                                        THEN 'TWITTER_DECK_APP'
                                WHEN source_tweet LIKE '%>Twitter for iPad<%'
                                        THEN 'IPAD'
                                WHEN source_tweet LIKE '%>Twitter Web App<%'
                                        THEN 'TWITTER_WEB_APP'
                                WHEN source_tweet LIKE '%>EcuaBot%'
                                        THEN 'ECUABOTS'
                                WHEN source_tweet LIKE '%>Twitter Web Client%'
                                        THEN 'TWITTER_WEB_CLIENT'
                                WHEN source_tweet LIKE '%>Hootsuite Inc.<%'
                                        THEN 'HOOTSUITE'
                                WHEN source_tweet LIKE '%>IFTTT<%'
                                        THEN 'IFTTT'
                                WHEN source_tweet LIKE '%>TweetCaster for Android<%'
                                        THEN 'TWEETCASTER_ANDROID'
                                WHEN source_tweet LIKE '%>TweetCaster for iPhone<%'
                                        THEN 'TWEETCASTER_IPHONE'
```

```
                                WHEN source_tweet LIKE '%>elyex%'
                                    THEN 'ELYEX'
                                WHEN source_tweet LIKE '%>Tweetbot for i%'
                                    THEN 'TWEETBOOT_IOS'
                                WHEN source_tweet LIKE '%>Tweetbot for Windows<%'
                                    THEN 'TWEETBOOT_WINDOWS'
                                WHEN source_tweet LIKE '%>Tweetbot for Android<%'
                                    THEN 'TWEETBOOT_ANDROID'
                                WHEN source_tweet LIKE '%>Postcron App<%'
                                    THEN 'POSTCRON_APP'
                                WHEN source_tweet LIKE '%>Buffer<%'
                                    THEN 'BUFFER'
                                WHEN source_tweet LIKE '%>Periscope<%'
                                    THEN 'PERSICOPE_LIVE_VIDEO'
                                WHEN source_tweet LIKE '%>Pypbot<%'
                                    THEN 'PYPBOT'
                                WHEN source_tweet LIKE '%>Mobile Web%'
                                    THEN 'WEB_MOBILE'
                                WHEN source_tweet LIKE '%>Echofon<%'
                                    THEN 'ECHOFON'
                                WHEN source_tweet LIKE '%>Sprout Social<%'
                                    THEN 'SPROUT_SOCIAL'
                                WHEN source_tweet LIKE '%>twitter bot%'
                                    THEN 'TWITTER_BOT'
                                WHEN source_tweet LIKE '%>UberSocial for Android%'
                                    THEN 'UBERSOCIAL_ANDROID'
                                WHEN source_tweet LIKE '%>UberSocial for iPad%'
                                    THEN 'UBERSOCIAL_IPAD'
                                WHEN source_tweet LIKE '%>UberSocial for iPhone%'
                                    THEN 'UBERSOCIAL_IPHONE'
                                ELSE 'OTRO'
                        END AS source_tweet_classified
                        FROM transporte
                ) AS TWEET_INFO
        INNER JOIN FECHAS_TRANSPORTE FA ON FA.TWEET_CLASIFIED_ID = id_tweet
GO
```

ANEXO 4

SCRIPT R PARA ENCONTRAR COINCIDENCIAS DE BARRIOS Y PARROQUIAS

```
source <- cbind(data.frame( NOMBRE_BARRIO = sapply(docsSource, as.character),
stringsAsFactors = FALSE), source)

coincidencias <- data.frame(d[unlist(source$NOMBRE_BARRIO),1:2])
coincidencias$NOMBRE_BARRIO <- coincidencias$word
coincidencias <- aggregate(coincidencias$freq, by= list(NOMBRE_BARRIO =
coincidencias$NOMBRE_BARRIO), FUN= max)
coincidencias <- na.omit(coincidencias)
  coincidencias <- subset(coincidencias,
    coincidencias$NOMBRE_BARRIO != 'NA'
    & coincidencias$NOMBRE_BARRIO != ''
    & coincidencias$x !='NA'
  )
  resultado <- coincidencias[order(-coincidencias$x),]
```

ANEXO 5

SCRIPT R PARA ENCONTRAR COINCIDENCIAS DE PARADAS

```
#COMPRAR COINCIDENCIAS DE PARADAS DE BUSES EN TRANSPORTE
source <- cbind(data.frame(NOMBRE_EDITADO = sapply(docsSource, as.character),
stringsAsFactors = FALSE),source)
  resultado <- data.frame(ANALISIS =
d[unlist(unique(source$NOMBRE_EDITADO)),1:2])
  resultado$NOMBRE_EDITADO <- resultado$ANALISIS.word
  resultado$COUNT <- resultado$ANALISIS.freq
  resultado <- resultado[order(-resultado$ANALISIS.freq),]
  return(na.omit(resultado[1:25,]))


#COMPARAR COINCIDENCIAS DE PARADAS DEL SISTEMA INTEGRADO EN TRANSPORTE
  source <- cbind(data.frame(NOMBRE_EDITADO = sapply(docsSource,
as.character), stringsAsFactors = FALSE),source)
  resultado <- data.frame(ANALISIS = d[unlist(source$NOMBRE_EDITADO),1:2])
  resultado$NOMBRE_EDITADO <- resultado$ANALISIS.word
  resultado$COUNT <- resultado$ANALISIS.freq
  resultadoResumido <- aggregate(resultado$COUNT, by= list(NOMBRE_EDITADO =
resultado$NOMBRE_EDITADO), FUN = max)
return(resultadoResumido)
```