ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

MINERÍA DE TEXTO DE LA WEB, DE OPINIÓN PÚBLICA Y HECHOS REFERENTES AL BARRIO LA FLORESTA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

VARGAS PULLIQUITÍN MARLON FABRICIO

marlon.f.vargas.p@outlook.com

DIRECTORA: ING. ELISA KARINA MENA MALDONADO MSC.

elisa.mena@epn.edu.ec

CODIRECTOR: ING. IVÁN MARCELO CARRERA IZURIETA MSC.

ivan.carrera@epn.edu.ec

Quito, junio 2018

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Marlon Fabricio Vargas Pulliquitín, bajo nuestra supervisión.

Ing. Elisa Karina Mena Maldonado MSc.

DIRECTORA DE PROYECTO

Ing. Iván Marcelo Carrera Izurieta MSc.

CODIRECTOR DE PROYECTO

DECLARACIÓN

Yo Marlon Fabricio Vargas Pulliquitín, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Marlon Fabricio Vargas Pulliquitín

AGRADECIMIENTOS

Agradezco a Fabricio Vargas y Janeth Pulliquitín, por haberme educado con todo su amor y dedicación. A Jeanneth Vargas y Samy Vargas, por ser siempre mi apoyo y fuente de alegría infinita. A Nestor Pulliquitín y Teresa Jurado, por transmitirme su sabiduría y cariño con mucha paciencia. A mi tía Azucena Jacho, por cuidarme y estar siempre pendiente de mí. Y a los demás miembros de mi familia con los que crecí, todos forman parte de la realización de esta meta.

Agradecimientos especiales a la Ing. Elisa Mena y al Ing. Iván Carrera, por su constancia y por ser mi guía a lo largo de la realización de este trabajo. Al Ing. Gustavo Samaniego, por sus sabios consejos. A la Facultad de Ingeniería de Sistemas y a la Escuela Politécnica Nacional, por formarme como profesional.

DEDICATORIA

Mis esfuerzos en alcanzar esta meta los dedico a Samy Vargas. Aunque no pude estar todo el tiempo ahí para verte crecer, te llevé conmigo siempre.

ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

ÍNDICE DE CÓDIGOS

RESUMEN

El presente proyecto detalla la construcción de un sistema de analítica de datos orientado a la minería de texto. El análisis de los datos se centrará en determinar los temas de los textos, los sentimientos y emociones que estos expresan. Este sistema se elabora mediante un enfoque de análisis descriptivo y predictivo.

Los datos recolectados provienen de fuentes web como sitios de noticias, blogs y redes sociales con respecto al barrio La Floresta, de Quito. El producto final de la ejecución del proyecto es un sistema que tiene la capacidad de clasificar texto según la temática a la que pertenece e identificar los sentimientos y las emociones expresadas en él. El sistema presenta los resultados del análisis de forma gráfica en una interfaz web.

Los resultados del procesamiento de los datos en el sistema nos permiten observar que, de acuerdo con las temáticas del texto y sus sentimientos, La Floresta es un barrio unido y organizado con una oferta cultural y artística abundante.

Palabras Clave: La Floresta, Minería de texto, Clasificación de datos, Análisis de sentimientos.

ABSTRACT

The present project details the construction of a data analytics system oriented to text mining. The data analysis focuses on determining the topics of the texts, with the feelings and emotions expressed in them. This system is elaborated through a descriptive and predictive analysis approach.

The data collected was gathered from web sources such as news web sites, blogs and social media with about the neighborhood La Floresta of Quito. The final product of the execution of the project is a system that has the functionality of classifying text according to the topic to which it belongs to, and identifying the feelings and emotions expressed in it. The system presents the results of the analysis graphically in a web interface.

The results of the processing of data in the system show us that, according to the topics and the feelings of the texts, La Floresta is a united and organized neighborhood with a great cultural and artistic offer.

Key words: La Floresta, Text mining, Data classification, Sentiment analysis.

1. INTRODUCCIÓN

1.1 Motivación

Actualmente, se estima que la cantidad de datos alojados en Internet se duplica cada dos años; de dichas cantidades, sólo el 25% es considerado útil para el análisis. Sin embargo, aproximadamente sólo el 3% es analizado (Gantz & Reinsel, 2012). Gran parte de estos datos están disponibles para adquirirlos y analizarlos en busca de ideas de provecho.

Los medios sociales y de noticias son sitios donde se publica constantemente texto de opinión pública y narración de hechos. Estos sitios son fuentes de datos que pueden ser utilizados para averiguar de qué temas se habla y en qué cantidad, si se habla positiva o negativamente, o qué emociones expresan o causan en sus lectores; todo a través de un proceso de minería de texto en el cual se analizan datos textuales para descubrir patrones y tendencias de interés (Liu, 2015). Al averiguar sobre qué temáticas se comentan, los sentimientos y emociones que causan estas en las personas, se pueden estimar sus necesidades. A su vez, “cada necesidad insatisfecha representa una oportunidad de negocio” (Strategyn, 2017).

Respecto a este tema, se realizó un proyecto en el año 2016 en Corea, en el que se utilizó minería de texto sobre posts de Twitter donde mencionaban a pequeños negocios locales de comida. El objetivo fue implementar un sistema de descubrimiento automático de conocimiento a partir de comentarios publicados en la red social; esto para determinar el nivel de satisfacción de los clientes y ofrecer a dichas empresas una herramienta para la toma de decisiones. Se utilizaron métodos como el *web crawling* y el consumo de servicios a través de *Application Programming Interfaces* (APIs) para la adquisición de datos, junto a técnicas de aprendizaje de máquina para el análisis. Como resultados, cuantificaron la conformidad que expresaban las opiniones vertidas, clasificándolas por negocio e identificando perfiles de su clientela según edad, sexo y ocupación (Sung-min & Sung-min, 2016).

Otro proyecto similar se realizó en México. El trabajo se basó en recolectar publicaciones en Twitter de la prensa chilena, en búsqueda de titulares que hagan alusión a México o al pueblo mexicano. Entre los principales objetivos, estaba el descubrir la imagen que proyecta el país centroamericano a los demás países. Los resultados mostraron que, de

todas las publicaciones, el 0,5 % se referían a México, y se encontró que los tres temas más abordados fueron: en primer lugar, el fútbol, en segundo lugar, la delincuencia y otros delitos en México, y en tercer lugar, noticias sobre artistas mexicanos (Cárcamo, Calva, Ronquillo, & Nesbet, 2017).

De observar estos hechos, surge el desarrollo del presente trabajo, el cual abarca el análisis de texto tomado de la web que contenga opinión pública y hechos referentes al barrio La Floresta. Este barrio se encuentra en Quito - Ecuador, en la parroquia La Mariscal. Esta parroquia y el Centro Histórico son consideradas las zonas especiales turísticas de Quito (Quito Turismo, 2013). La Floresta es un barrio que despierta el interés turístico y alberga una alta actividad económica; está en el puesto número 14 con mayor población económicamente activa, de entre 147 barrios pertenecientes a la administración zonal Eugenio Espejo (Gobierno Abierto Quito, 2017); alrededor de 14.500 contribuyentes al Servicio de Rentas Internas, entre personas naturales y sociedades, residen o realizan alguna actividad económica en el barrio (Servicio de Rentas Internas, 2017). También posee una alta concentración de negocios propios de los moradores del barrio. Tiene alrededor de 93 negocios dentro de un área de aproximadamente 1,1 km² (De La Floresta, 2016) (Secretaria de Territorio, Habitat y Vivienda - Municipio de Quito, 2013). El Colectivo cultural barrial “De La Floresta”, conformado con el propósito de unir y organizar a los vecinos y emprendedores del barrio, realiza eventos para promover el consumo de sus productos y servicios (De La Floresta, 2016).

Los pequeños negocios locales del barrio han sido montados basándose en la experiencia y conocimiento empírico de sus propietarios (Moshenek, 2017). En la web se encuentran diversas publicaciones referentes al barrio, realizadas por residentes, negociantes, visitantes nacionales e internacionales, y sitios de noticias (De La Floresta, 2016) (The Culture Trip, 2017). Sin embargo, su uso para la toma de decisiones es dificultoso debido a la falta de organización, procesamiento e interpretación de los datos.

1.2 Iniciativa

El presente proyecto pretende desarrollar un sistema automatizado de minería de texto, que permita determinar cuánto y sobre qué temáticas se habla dentro y fuera del barrio; así como averiguar qué sentimientos se expresan en dichas publicaciones. El sistema podrá servir de herramienta a los moradores, emprendedores, negociantes, inversionistas, y cualquier persona interesada en conocer las necesidades de la gente que habita o frecuenta el barrio La Floresta, y que requiera hacer uso de esta información con motivos de negocio o para idear iniciativas de mejoramiento en cualquier aspecto del barrio.

Se propone la construcción de un sistema con la capacidad de adquirir y analizar datos de redes sociales, sitios de noticias y blogs referentes al barrio La Floresta. Al finalizar la adquisición y el análisis de los datos, se mostrarán los resultados mediante un tablero de mando.

La implementación del presente proyecto se basa en la metodología propuesta por Erl, Khattak & Buhler (2016) para las fases de un proyecto de analítica de datos. Así, las actividades realizadas fueron:

1. Identificación de datos
2. Adquisición y filtrado de datos
3. Extracción de datos
4. Validación y limpieza de datos
5. Agregación y representación de datos
6. Análisis de datos (Minería de texto)
7. Visualización de datos

Se identificaron como fuentes de datos: Facebook, Twitter, El Comercio, El Telégrafo, TripAdvisor y The Culture Trip, debido a su popularidad y a que éstas poseen la mayor cantidad de publicaciones referentes a La Floresta. De estas fuentes, se recogen datos referentes al barrio, mediante un proceso de adquisición y filtrado. Antes de realizar el análisis, se realizan tareas de preprocesamiento como la extracción, validación y limpieza. Esto permite tener datos válidos, limpios y en formatos procesables.

Debido a que los datos a manejar son exclusivamente de texto, la fase de análisis corresponde concretamente a minería de texto. Se propone realizar un análisis de tipo exploratorio y descriptivo, sin partir de una hipótesis, sino examinando los datos en

búsqueda de patrones o anomalías. El análisis debe tener la capacidad de responder preguntas sobre eventos pasados o presentes (Erl, Khattak, & Buhler, 2016). Los resultados finales del proyecto son un sistema automático de analítica de texto, y un tablero de mando, como herramienta web de visualización de los resultados del análisis.

El desarrollo del proyecto parte de las siguientes afirmaciones:

- El proyecto se desarrolló con fines académicos y sin ánimos de lucro,
- La adquisición de datos realizada no es sancionada por la ley. Los textos recogidos están en sitios web de acceso público y los objetivos del proyecto no van en contra de los intereses de los propietarios de las fuentes. Además, se hace mención de las fuentes de datos para dar el crédito correspondiente.
- En el desarrollo del sistema, se utilizan modelos y librerías preexistentes para el análisis, no se pretende mejorar o desarrollar nuevos.

2. METODOLOGÍA

El presente proyecto realiza una investigación de tipo aplicada y exploratoria. Se refiere a una investigación aplicada, porque busca solucionar un problema real y específico basándose en teorías y fundamentos prestablecidos. Y se refiere también a una investigación de tipo exploratoria, porque no parte de una hipótesis inicial, sino que se pretende recabar datos en busca de patrones de interés (Posso, 2011).

La metodología utilizada se basa en el ciclo de vida de analítica de datos propuesto por Erl, Khattak, & Buhler (2016) en su libro *Big Data Fundamentals*. *Big Data* es el campo multidisciplinario dedicado al análisis de grandes volúmenes de datos no estructurados[1]. Las características de los datos en un proyecto de *Big Data* son: volumen, variedad, velocidad, veracidad y valor. Juntas, las tres primeras características, definen si un proyecto corresponde o no a un caso de *Big Data*, mientras que las dos últimas dependen de la correcta ejecución de tareas como el filtrado, limpieza, y la validación de los datos (Erl, Khattak, & Buhler, 2016). En base a esta afirmación, y al hecho de que los datos tratados en este proyecto presentan solamente variedad, una de las tres de las características excluyentes para ser *Big Data*, el presente es un caso de analítica de datos, pero no de Big Data. Aun así, debido a la necesidad de procesar datos no estructurados, la metodología de analítica de Erl, Khattak, & Buhler (2016) se adapta perfectamente a las necesidades del presente proyecto.

Basándose en las fases de un proyecto de analítica enfocada al caso de minería de texto web, de opinión pública y hechos, se diseñó la arquitectura del sistema que se muestra en la Ilustración 1.

El proceso empieza con la identificación de fuentes web de opinión pública y noticias, como Facebook, Twitter, Trip Advisor, El Comercio, etc., de ellos se recolecta el texto mediante un proceso de adquisición y filtrado, usando R y Python. Después de la recolección, los datos son almacenados en bases de datos en CouchDB, por motivos de registro y para

[1] Los datos no estructurados son aquellos datos sin estructuras bien definidas como las que se tienen tradicionalmente en las bases de datos relacionales (Erl, Khattak, & Buhler, 2016).

otros posibles análisis futuros. De la recolección, los datos entran al escenario de preprocesamiento de datos, donde, a través del uso de R, se realiza extracción, validación y limpieza de los datos para su posterior almacenamiento en un nuevo conjunto de bases de datos limpios, en CouchDB; estas tareas preparan los datos para el análisis.

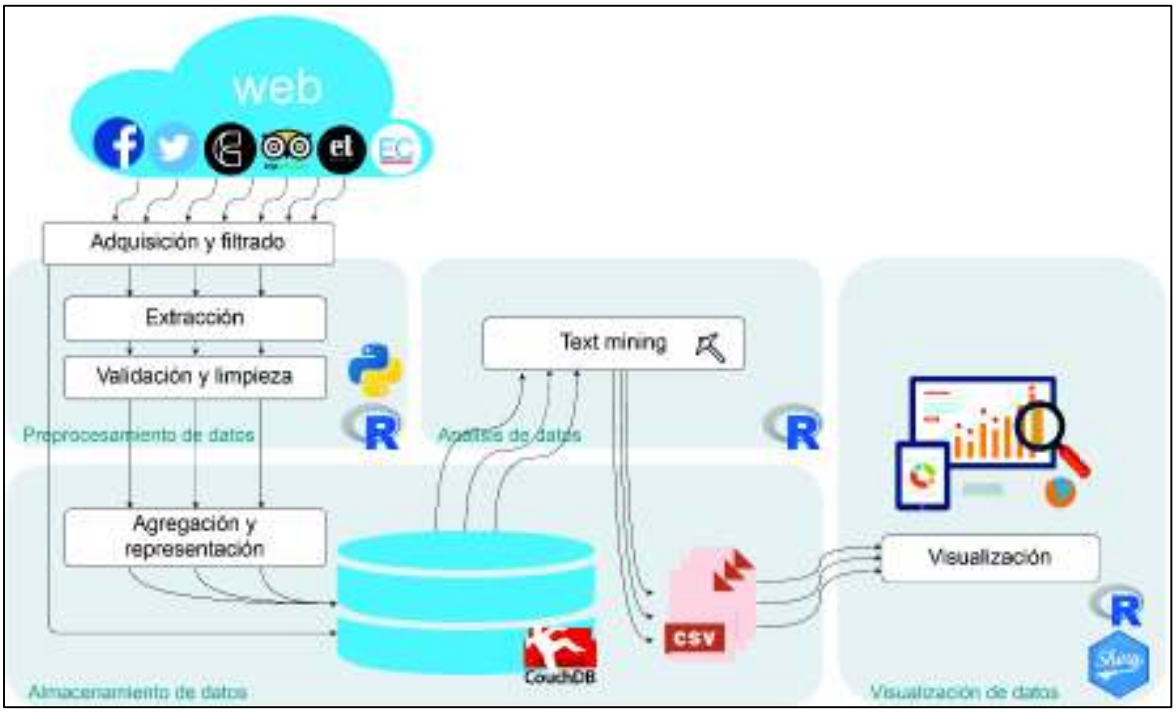


Ilustración 1. Arquitectura del sistema de minería de texto web

El siguiente paso es llevar los datos a la fase de análisis, donde, utilizando R, se aplican varias técnicas de clasificación y descripción de datos. Los resultados son almacenados en archivos de texto plano en formato CSV, por su fácil lectura y almacenamiento. Estos archivos son cargados al escenario de visualización, donde son leídos con R y presentados a través de gráficos interactivos mediante una aplicación web, construida y publicada usando Shiny, de R.

En las siguientes secciones, 2.1.Identificación de datos, 2.2.Adquisición y filtrado de datos, 2.3.Extracción de datos, 2.4.Validación y limpieza de datos, 2.5.Agregación y representación de datos, 2.6.Análisis de datos (Minería de texto), 2.7.Visualización de datos, se detalla el funcionamiento de cada una de las etapas del proyecto.

2.1 Identificación de datos

Es el proceso de identificar el conjunto de datos necesarios y sus fuentes. Según el alcance del proyecto, las fuentes pueden ser internas, como almacenes de datos, o externas a la organización, como proveedores de datos u otras formas, como blogs o sitios web basados en contenidos. La identificación de varios tipos de fuentes de datos amplía la probabilidad de encontrar patrones escondidos (Erl, Khattak, & Buhler, 2016)

El presente proyecto busca analizar datos de opinión pública y hechos encontrados en la web. La selección de fuentes de datos se realizó con una búsqueda en internet, identificando las fuentes con mayor número de entradas sobre el tema, y en base a su popularidad y veracidad.

Como fuentes de datos de opinión en redes sociales se seleccionaron Facebook y Twitter. Ambas son calificadas como las dos redes sociales más utilizadas en el país según Interactive Advertising Bureau (IAB) Ecuador (2017). El contenido de Twitter es de acceso público, mientras que el de Facebook no lo es. Facebook tiene rigurosas restricciones de acceso a los datos de sus usuarios por motivos de privacidad. Sin embargo, las páginas de Facebook publicadas y sus contenidos eran públicos, hasta finales de marzo de 2018, y se podía acceder a ellos si se los tenía identificados. Esto cambió debido a una actualización de las políticas de seguridad de Facebook, y ahora se exige la adquisición de permisos especiales para lograr el acceso a los datos de páginas públicas. Facebook tampoco permite, a través de su API oficial, realizar búsquedas automatizadas de sus páginas o sus usuarios. Por esta razón, mediante la herramienta de búsqueda manual de Facebook, se seleccionaron los cuatro resultados más relevantes de páginas referentes al barrio La Floresta. Estas páginas son: Barrio La Floresta – Quito, De La Floresta, La Floresta te Enamora y Te Quiero Verde La Floresta (Facebook, 2017).

Los sitios web de viajes que ofrecen servicios de reservación e información sobre hoteles, restaurantes y sitios por visitar, también contienen reseñas realizadas por sus usuarios; esto las convierte en una fuente importante de opiniones de visitantes nacionales y extranjeros. De dichos sitios, se escogió a *Trip Advisor* por poseer la más fuerte comunidad de reseñas generadas por usuarios (Chen C. , 2017). Otra fuente, que brinda una visión desde el punto de vista turístico, es The Culture Trip, un sitio web tipo blog, donde los viajeros cuentan sus historias y experiencias después de visitar un destino. *The Culture*

Trip tiene 12,5 millones de seguidores y aportantes de contenido, y muestra del interés que poseen turistas extranjeros en este barrio de Quito (Forbes, 2018) (The Culture Trip, 2018).

Las fuentes de datos de hechos corresponden a sitios web de noticias pertenecientes a dos diarios de amplia difusión en Ecuador, con el mayor número de entradas referentes al barrio La Floresta de Quito; el diario El Comercio, con 63 entradas hasta el 6 de enero del 2018 (El Comercio, 2018) y El Telégrafo con 40 hasta la misma fecha (El Telégrafo, 2018).

El esquema de las fuentes seleccionadas agrupadas por tipo se muestra en la siguiente Ilustración 2.

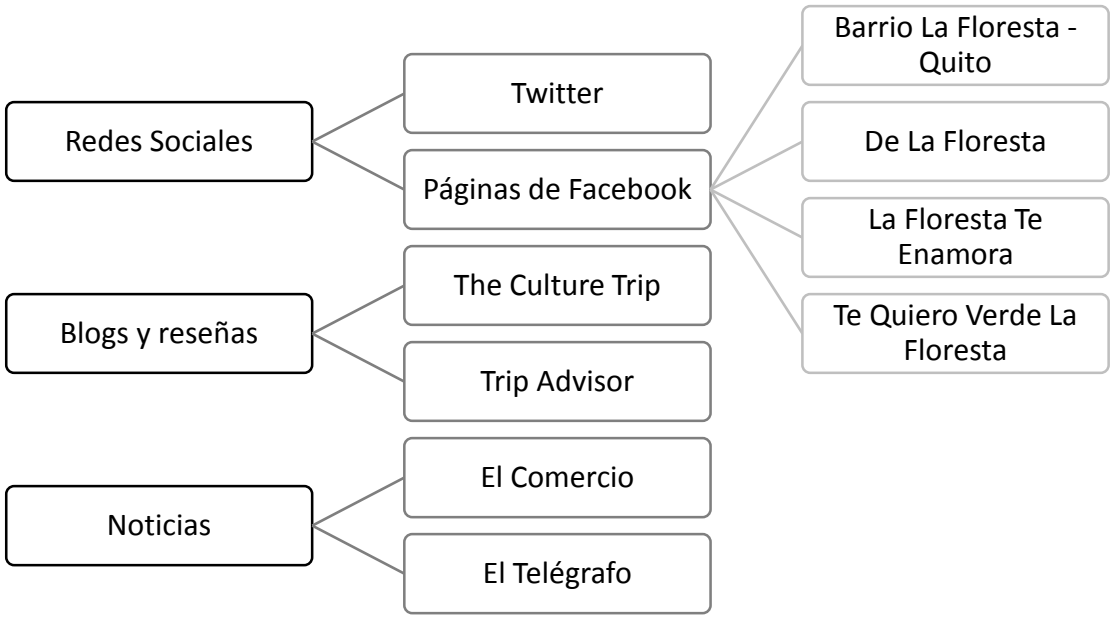


Ilustración 2. Esquema de fuentes de datos seleccionadas según su tipo

2.2 Adquisición y filtrado de datos

Durante la etapa de adquisición y filtrado de datos, estos se recolectan de todas las fuentes identificadas en la fase de 2.1.Identificación de datos. Es probable que los conjuntos de datos adquiridos traigan consigo datos que no son de interés para el análisis, por lo que es preciso realizar operaciones de filtrado y preprocesamiento (Chen, Mao, & Liu, 2014).

Para la recolección de datos web, se utilizaron principalmente dos técnicas: *web scraping*, basado en la descarga e interpretación de páginas web para recolectar datos específicos como títulos, subtítulos, descripciones u otras zonas específicas de un sitio (Picot, 2016); y el consumo de datos a través de APIs, que son interfaces a través de los cuales las aplicaciones solicitan y comparten datos (Kwartler, 2017).

Los datos pueden presentarse en distintas formas. Pueden ser estructurados, semiestructurados o no estructurados. Los datos estructurados poseen formatos, tipos de datos y estructuras bien definidas. Los datos semi-estructurados corresponden a datos textuales de patrones deducibles que hacen posible su interpretación gramática, como *JavaScript Object Notation* (JSON), *eXtensible Markup Language* (XML) o *HyperText Markup Language* (HTML). Y los datos no estructurados son aquellos que no poseen una estructura definida, como documentos de texto, archivos PDF, y archivos multimedia (Wiley, 2015).

Para el presente proyecto se recogió la mayor cantidad de datos históricos posibles encontrados en las fuentes. Los datos más antiguos son de 2011.

El filtrado de datos se realizó junto con la adquisición, en la configuración de los criterios de búsqueda aplicados en las fuentes. Los datos adquiridos y filtrados son almacenados en bases de datos, utilizando el gestor CouchDB. Los datos sin procesar son almacenados por motivos de respaldo, y en la práctica, las organizaciones almacenan estos datos para otros tipos de análisis que se pueden realizar en el futuro (Erl, Khattak, & Buhler, 2016).

De acuerdo con la fuente de datos, se utilizaron técnicas con las que se adquirieron los datos y se aplicaron los filtros. Así, las fuentes de datos fueron: Twitter, Facebook, The Culture Trip, Trip Advisor, El Comercio y El Telégrafo.

2.2.1 Twitter

En un inicio se planeó utilizar la API oficial de Twitter para la recolección de datos, ya que este método es efectivo para la captura de datos en tiempos cercanos al real. Sin embargo, la API no ofrece una función para capturar *tweets* realizados antes de diez días, a menos que se adquiera una suscripción pagada con funcionalidades adicionales y mejoradas (Twitter Developers, 2017).

Al ser necesaria la recolección de *tweets* históricos, se encontró una librería abierta que permite este trabajo. *GetOldTweets*, una librería disponible para Python y Java (Henrique, 2018). El mecanismo detrás de *GetOldTweets* es un web scraper orientado a obtener datos históricos de Twitter (Henrique, 2018).

En Código 1, se muestra el uso de *GetOldTweets* para recoger tres *tweets* mediante el patrón de búsqueda “La Floresta”. El ejemplo es una versión resumida del programa que adquiere los *tweets* dentro del sistema. Como se puede observar, la librería permite una adquisición de *tweets* mediante la configuración de varios criterios de búsqueda en la función *TweetCriteria*, que a la vez hacen el trabajo de filtrar los datos a recoger. Dentro de esta función se define: en *setSince*, una fecha mínima de los *tweets* a recoger (variable *since*); en *setUntil*, la fecha máxima que para este caso es la fecha actual (variable *until*); en *setNear*, un punto de ubicación geográfica de referencia, el cual, a través de Google Maps, se tomó como el centro aproximado del barrio La Floresta (variable *point*); en *setWithin*, un radio dentro del cual se realizaron las publicaciones (variable *radio*); en *setQuerySearch*, el patrón de texto a buscar (variable *query*); y en *setMaxTweets*, el número máximo de *tweets* a recoger (variable *max_tweets*).

```
# Archivo: get_tw.py
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# importar librería
import got
# Definir criterios de búsqueda
since  = "2014-01-01"
until = (datetime.now() + timedelta(days=1)).strftime("%Y-%m-%d")
point = "-0.202689,-78.496300"
radio = "15mi"
query = "La Floresta"
max_tweets = 3
tweetCriteria = got.manager.TweetCriteria()
                    .setSince(since)
                    .setUntil(until)
                    .setNear(point)
```

```
                    .setWithin(radio)
                    .setQuerySearch(query)
                    .setMaxTweets(max_tweets)
i = 0
while True:
    # Obtener tweets
    tweet = got.manager.TweetManager.getTweets(tweetCriteria)[i]

    # Imprimir tweets
    print('\n Tweet encontrado:')
    print('id: '+ str(tweet.id))
    print('user: '+ str(tweet.username))
    print('text: '+ str(tweet.text))
    print('date: '+ str(tweet.date))
    i += 1
```

Código 1. Ejemplo de programa en Python para recolectar tweets

La ejecución del programa recoge los *tweets* y los despliega como se muestra en la Ilustración 3.

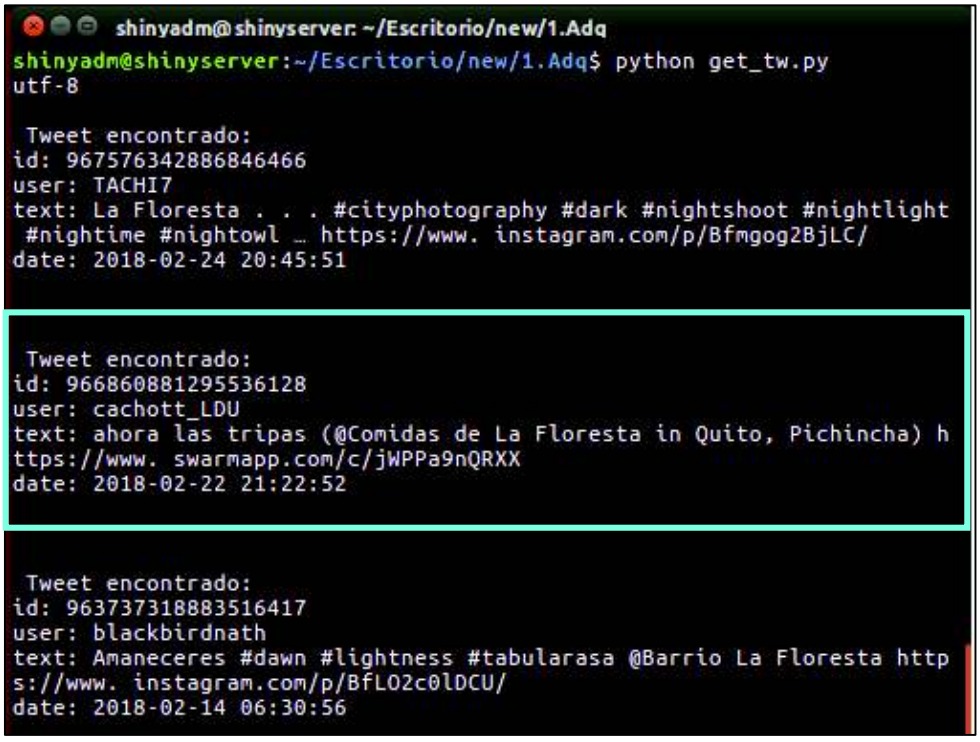


Ilustración 3. Resultado de la obtención de tweets

Como se observa en la Ilustración 3, se obtuvieron tres *tweets*, el segundo corresponde al tweet de la Ilustración 4. Se extrajo el texto del tweet (*text*), el identificador de la publicación

(*id*), el nombre de usuario (*user*) y la fecha (*date*). El tweet original se muestra en la Ilustración 4.

Ilustración 4. Ejemplo de tweet acerca de La Floresta

Una de las ventajas de *GetOldTweets* es que el acceso a los atributos de los *tweets* es muy práctico; no sólo nos permite adquirir *tweets*, sino que los devuelve en forma semi-estructurada. Otra de las ventajas de utilizar esta librería, es que prescinde el uso de un *token* de acceso. Un token de acceso es una medida de seguridad de validez temporal, para proteger el acceso de entidades no autorizadas. Por otro lado, la ventaja de la API oficial de Twitter, son sus cortos tiempos de respuesta versus el tiempo que le toma a *GetOldTweets* realizar *web scraping.*

2.2.2 Facebook

Para la adquisición de posts de Facebook se utilizó su API oficial. Esta herramienta permite acceder a los datos de páginas, usuarios, publicaciones, grupos y eventos (Facebook, 2018), mediante un servicio que puede consumirse a través de peticiones http, en este caso, generadas por un programa escrito en Python. La librería *Facebook* para Python se encarga del manejo de las peticiones necesarias para conseguir los datos. Facebook requiere también de un token de acceso temporal.

Para recolectar los posts de Facebook, se puede utilizar su API con el siguiente Código 2:

```
# Archivo: get_fb.py
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Importar librería
import facebook

# Definir token de acceso
token = 'EAAbCIWXsqqIBA...'

# Definir id de página de donde adquirir posts
page = '249188241799172'

# Conectarse a API de Facebook
graph = facebook.GraphAPI(token)

# Definir atributos deseados de los posts
all_fields = [
    'id',
    'message',
    'created_time',
    'likes.summary(true)',
    'comments.summary(true)',
    ]
all_fields = ','.join(all_fields)

# Obtener posts
posts = graph.get_connections(page, 'posts', fields = all_fields)

# Imprimir posts
for post in posts['data']:
    print '\nPost encontrado'
    print json.dumps(post, indent=4, sort_keys=True)
```

Código 2. Ejemplo de programa en Python que extrae posts de página pública de Facebook

En el Código 2, se observa que de antemano se deben tener dos cosas: el token de acceso y la página identificada mediante su *id*, de la que queremos adquirir sus posts. Una vez definido esto, se establece la conexión con la API, haciendo uso del token de acceso. Se definen los campos que requerimos del post, y los enviamos junto con el identificador de la página y el tipo de dato que queremos recoger dentro de la función *get_connections*. El resultado de esta función es un arreglo de objetos tipo JSON, en este caso, correspondientes a los posts encontrados en la página.

Al imprimir los resultados, se tiene lo mostrado en la Ilustración 5.

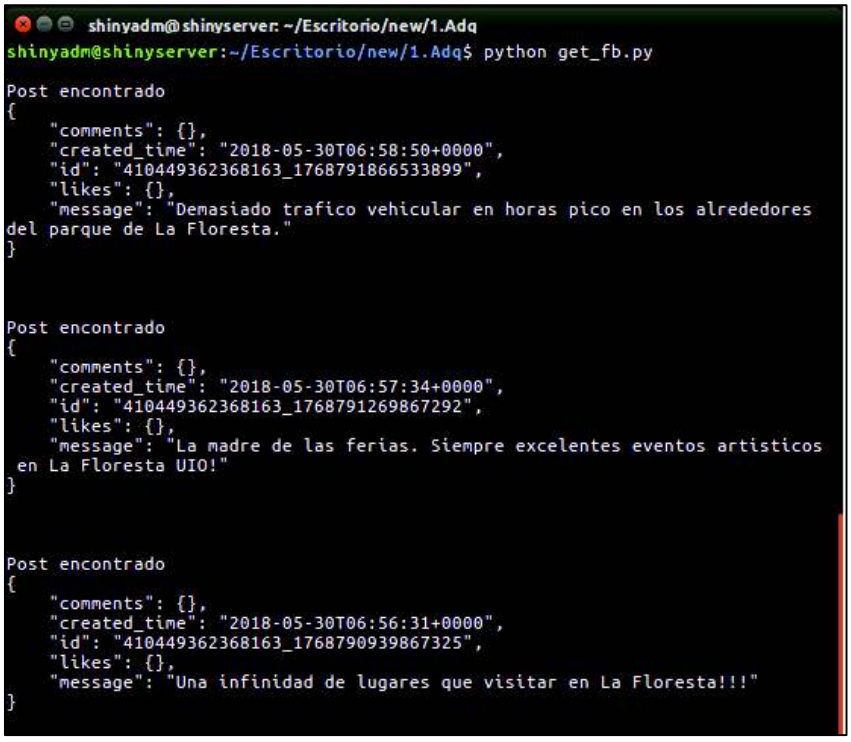


Ilustración 5. Posts de Facebook recolectados con la API

Los resultados indicados en la Ilustración 5 muestran que los datos entregados por la API de Facebook llegan en un formato tipo JSON distinto de respuesta de *GetOldTweets*. Esto nos da indicios de la necesidad de estandarizar el formato de los registros, en la siguiente fase del proyecto, 2.3.Extracción de datos.

Ilustración 6. Posts de página pública de Facebook

Los posts recogidos de la página pública se muestran en la Ilustración 5.

2.2.3 The Culture Trip

Para la recolección de datos desde The Culture Trip, se utilizó la técnica de *web scraping* y el buscador propio del sitio para su filtrado. En esta fuente, se comenzó por emular una búsqueda manual del patrón “La Floresta”. La búsqueda manual devolvió los resultados como se muestran en la Ilustración 7:

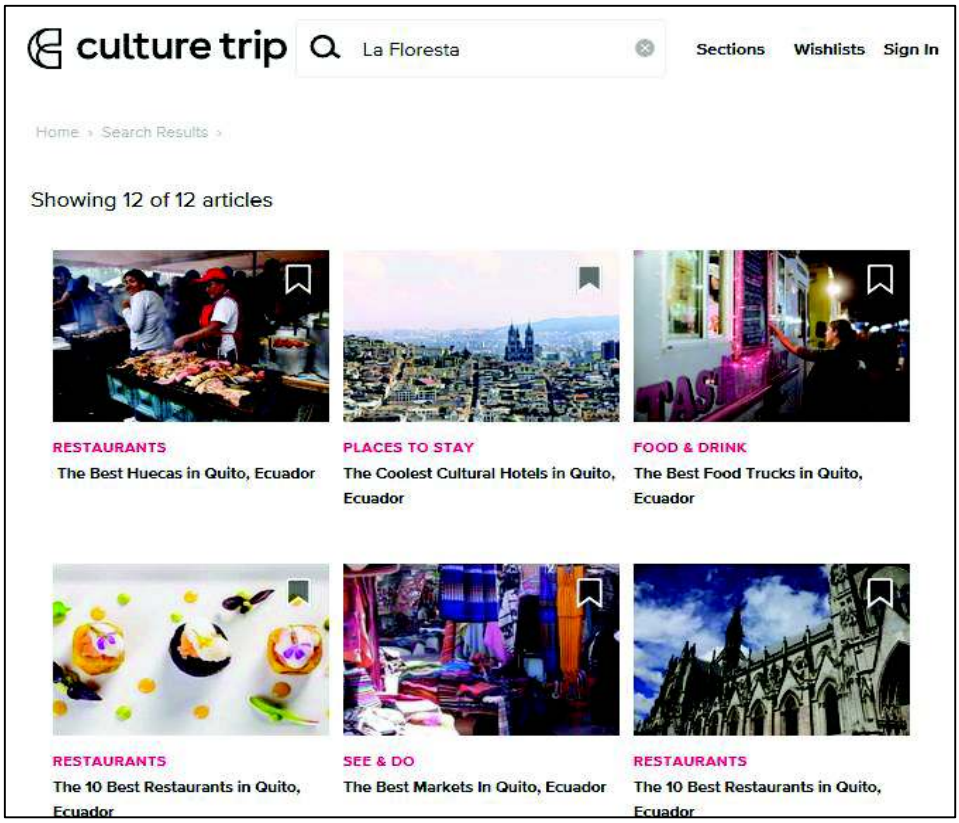


Ilustración 7. Resultados de búsqueda “La Floresta” (The Culture Trip, 2017)

El proceso de adquisición empezó, antes de usar *web scraping*, identificando una API que utiliza la página al realizar una búsqueda manual. Esta API permitió que la consulta de publicaciones pueda ser automatizada. Como se muestra en el Código 3, el criterio de búsqueda (variable *query*) es pasado como parámetro con el valor “La Floresta Quito Ecuador”, de esta forma se construye el enlace hacia la API. Mediante la función getURL, se solicitan los datos a la API y se almacenan (en la variable *out*) y se interpretan según su formato JSON con la función *fromJSON*.

```r
# Archivo: get_ct.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Definir criterio de búsqueda
query <- "La Floresta Quito Ecuador"

# Construir link hacia API de The Culture Trip
link <- paste0(
  'https://app.theculturetrip.com/cultureTrip-api/v1/articles?s=',
  gsub(" ","+",query),
  '&offset=0&limit=12'
  )
# Obtener datos de artículos
out <- getURL(link)
jsonData <- fromJSON(out)
```

Código 3. Obtención de datos de publicaciones de The Culture Trip

La respuesta obtenida (*jsonData*), como se muestra en la Ilustración 8, corresponde a un arreglo de objetos tipo JSON, donde cada uno lleva la información de una publicación del resultado de la búsqueda.

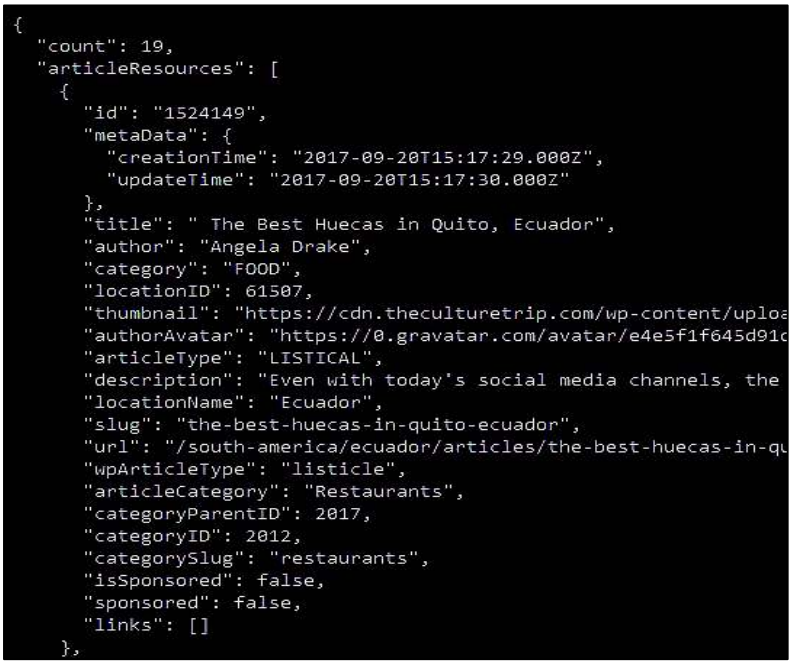


Ilustración 8. Extracto de respuesta de la búsqueda en The Culture Trip, usando la API

Una vez recibida la respuesta de la API, el objeto JSON es transformado a un formato de

dato tipo tabla, conocido como *dataframe*2, que contiene la mayoría de los atributos de cada artículo. Como se observa en la Tabla 1:

id	date	title	link
1524149	2017-09-20	Â The Best Huecas in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
1519238	2017-09-12	The Coolest Cultural Hotels in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
1490310	2017-08-25	The Best Food Trucks in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
396016	2015-09-11	The 10 Best Restaurants in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
393861	2015-08-11	The Best Markets In Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
114524	2014-03-04	The 10 Best Restaurants in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
1583962	2017-10-09	Essential Tips for Staying Safe in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
1508421	2017-09-05	The Top 7 Hiking Tour Operators in Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...
1408010	2017-07-25	The Best Day Trips from Quito, Ecuador	https://theculturetrip.com/south-america/ecuador/articl...

Tabla 1. Extracto tabla de artículos encontrados en The Culture Trip

Debido a que la API no proporciona el texto del artículo, fue necesaria la creación de una lista con los enlaces de todos los artículos encontrados, para acceder a cada uno de ellos usando un bucle que extrae el texto mediante web scraping.

La librería de lenguaje R utilizada para *web scraping* fue *rvest*. Rvest posee alrededor de 15 funciones, de las cuales, tres fueron las más utilizadas para el desarrollo (CRAN Project, 2016):

- read_html: Lee un archivo .html. Recibe como parámetro el URL de la página a leer.
- html_nodes: Identifica y extrae un determinado elemento (nodo) del HTML de la página. Recibe como parámetro el identificador del elemento (según XPath o selectores CSS).
- html_text: Extrae el contenido textual de un elemento del HTML de la página.

Usando su función *read_html*, se descarga el código de cada página web de los artículos. Después, mediante las funciones *html_nodes* y *html_text*, se accede a las secciones donde cada página aloja el contenido de texto, en este caso las etiquetas de párrafo (*p*), para irlo

[2] Un *dataframe*, en R, es un objeto de propiedades dimensionales similares a una matriz; con la diferencia de que un *dataframe* puede contener datos categóricos además de datos numéricos (Buechler, 2007).

recogiendo en una variable tipo texto. El extracto de la función que realiza el *web scraping* se muestra en el Código 4.

```
# Archivo: get_ct.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Leer página
webpage <- read_html(as.character(url))

# Extraer contenido
content <- paste(webpage %>%
                   html_nodes('p') %>%
                   html_text(),
                 collapse = " ")
```

Código 4. Función de web scraping en R

Al finalizar la recolección de texto, la tabla de artículos es completada con un campo adicional, que lleva el texto de su contenido correspondiente, como se observa en la Tabla 2.

id	date	title	link	text
114524	2014-03-04	The 10 Best Restaurants in Quito, Ecuador	https:/...	It might be hard to find this restaurant hidden behind p...
1352782	2017-06-09	Street Food You Have to Try in Quito, Ecuador	https:/...	To experience authentic street food in Quito, pay a visit t...
1408010	2017-07-25	The Best Day Trips from Quito, Ecuador	https:/...	Located a couple of hours away from Quito, Cotopaxi N...
1490310	2017-08-25	The Best Food Trucks in Quito, Ecuador	https:/...	Inka Burger serves some of the best hamburgers in Quit...
1508421	2017-09-05	The Top 7 Hiking Tour Operators in Quito, Ec...	https:/...	Topping the Trip Advisor list of best-rated hiking operat...
1519238	2017-09-12	The Coolest Cultural Hotels in Quito, Ecuador	https:/...	With only six double rooms, reservations are a must at t...
1524149	2017-09-20	Â The Best Huecas in Quito, Ecuador	https:/...	Bandera con sabor manabita <f0><U+009F><U+0098>...
1583962	2017-10-09	Essential Tips for Staying Safe in Quito, Ecua...	https:/...	Public transport is ridiculously cheap in Quito. It costs o...
393650	2015-08-11	10 Things To Do And See In Quito, Ecuador	https:/...	Guarding the historic streets of Quito's Old Town, The Vi...
393861	2015-08-11	The Best Markets In Quito, Ecuador	https:/...	This packed market in Quito's Mariscal region holds som...
396016	2015-09-11	The 10 Best Restaurants in Quito, Ecuador	https:/...	Sur, meaning "South", prides itself on high-quality food ...
406891	2015-10-15	The 10 Best Brunch Spots In Quito, Ecuador	https:/...	Jürgen Café is known for its delectable, and distinctly Ec...

Tabla 2. Extracto de tabla resultado de web scraping de artículos de The Culture Trip

Estos datos, en un formato semiestructurado, se encuentran almacenados en un dataframe alojado en memoria RAM.

2.2.4 Trip Advisor

Trip Advisor ofrece una guía completa de sitios turísticos de interés. Cada uno de estos posee una página informativa, en donde sus clientes comentan y califican diversos aspectos del lugar. La idea inicial fue recopilar todos los comentarios posteados acerca de

los sitios del sector de La Floresta, con el objetivo de analizar los sentimientos de cada uno y calcular un promedio agrupado por sitio de interés. Sin embargo, se notó que, junto con cada reseña, los clientes realizan una calificación de entre excelente, muy bueno, promedio, pobre y terrible. Esto se puede observar en la Ilustración 9.

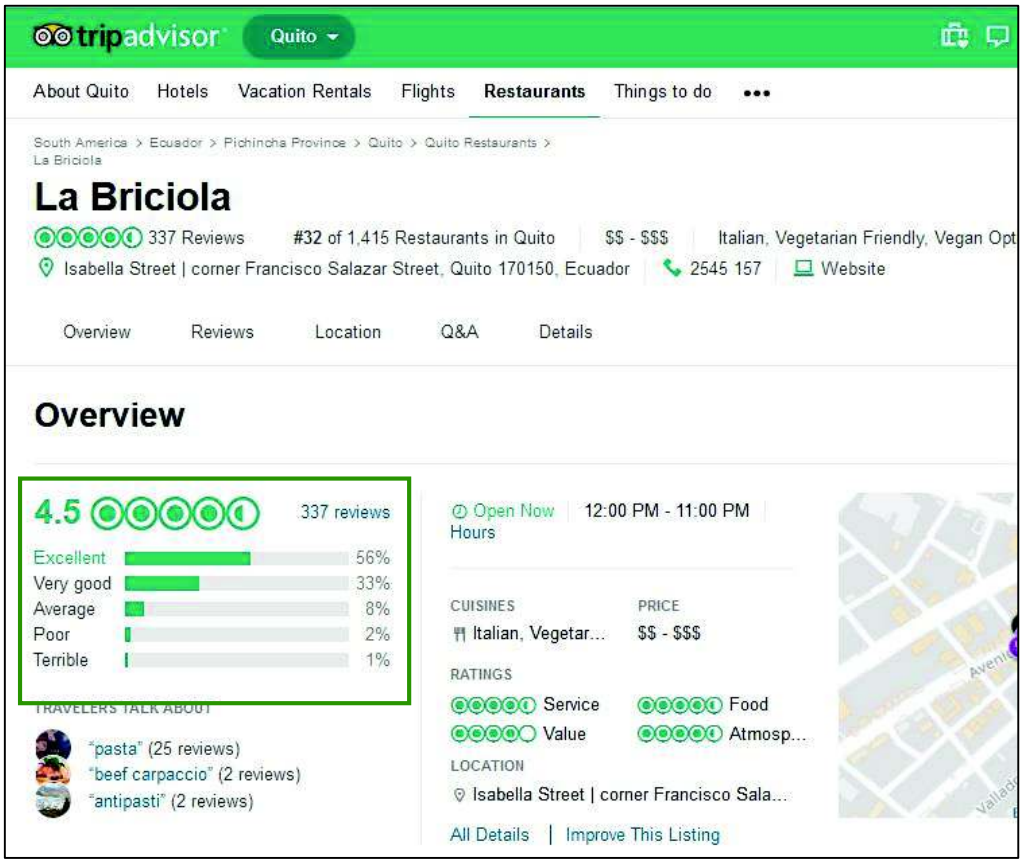


Ilustración 9. Ejemplo página de sitio de interés en La Floresta (Trip Advisor, 2017)

Con el propósito de tener una visión general de los sitios turísticos de La Floresta, se recogieron datos de *Trip Advisor* acerca de todos los lugares de interés de Quito: sus nombres, sus calificaciones, número de comentarios y su ubicación geográfica para desplegarlos en un mapa en la fase 2.7.Visualización de datos.

El primer paso fue identificar la API, de la que *Trip Advisor* obtiene todos los sitios turísticos de una zona determinada. El mapa, de la Ilustración 10, de *Trip Advisor* ubica los sitios de interés, que pueden ser hoteles y rentas (íconos azules y verdes), restaurantes (íconos morados) o cosas por hacer (en inglés *things to do*) (íconos anaranjados). Los sitios de interés con mayor número de reseñas se representan con un ícono con logo de color, mientras los de menor número de reseñas se representan solamente con puntos de color.

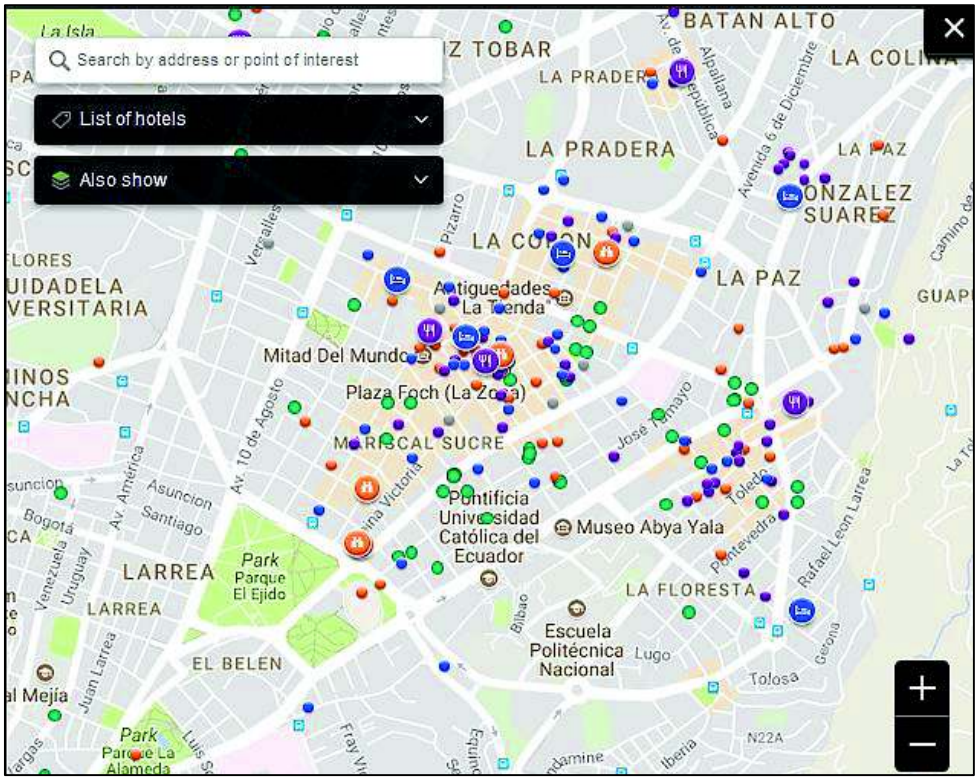


Ilustración 10. Mapa de sitios turísticos de La Mariscal (Trip Advisor, 2017)

Al recoger los sitios de toda la ciudad mediante la API, se seleccionaron solamente los campos de ubicación geográfica, enlace que lleva a la página web del sitio y el tipo, de hotel, restaurante, o cosas por hacer (que en general representan lugares de visita turística). Una vez construida una lista de los sitios con sus datos, se puede acceder, mediante un bucle y una función de web scraping, a la página web de cada sitio y recoger los otros tres datos faltantes: el nombre del lugar, la calificación y el número de comentarios posteados. Las funciones que recolectan estos datos se muestran en el Código 5.

Mediante una inspección de las páginas de los sitios, se ubicaron los elementos que se desean adquirir. Por ejemplo:

- Nombre del lugar: se ubicó en la etiqueta html de clase *heading_title*, como se observa en la Ilustración 11.

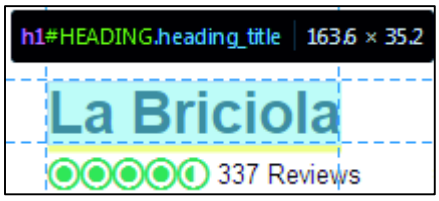


Ilustración 11. Identificación del elemento html, a recoger de Trip Advisor (Trip Advisor, 2017)

- Calificación: se ubicó en la etiqueta html de clase *overallRating*, como se observa en la Ilustración 12.

Ilustración 12.Identificación del elemento html, a recoger de Trip Advisor (Trip Advisor, 2017)

- Número de comentarios: se ubicó en la etiqueta html de clase rating, dentro de una etiqueta *a*, como se observa en la Ilustración 13.

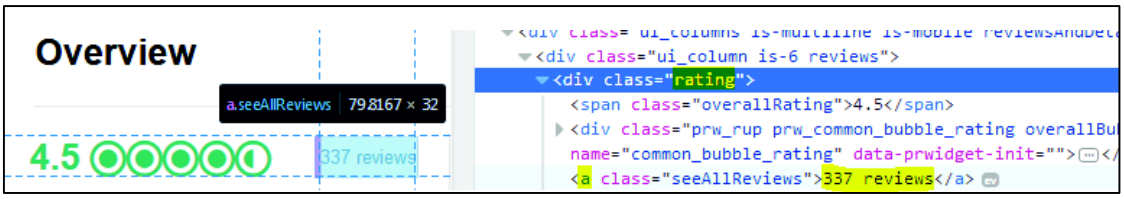


Ilustración 13.Identificación del elemento html, a recoger de Trip Advisor (Trip Advisor, 2017)

Después de identificar los elementos donde se encuentran los datos adicionales, éstos se recogen utilizando las funciones mostradas en Código 5.

```
# Archivo: get_ta.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Obtener nombre del lugar
place_name <- webpage %>%
  html_nodes('[class="heading_title"]') %>%
  html_text()

# Obtener calificación del lugar
rate <- webpage %>%
  html_nodes('[class="overallRating"]') %>%
  html_text()

# Obtener número de comentarios
num_comments <- webpage %>%
  html_nodes('[class="rating"]') %>%
  html_nodes('a') %>%
  html_text()
```

Código 5. Funciones de web scraping para recoger datos de sitios turísticos de Trip Advisor

Una vez recogidos todos los datos de los sitios y unificados en una sola tabla, el resultado se puede visualizar en la Tabla 3. Esta tabla tiene de campos la latitud (lat), longitud (lng),

la URL (*url*), el tipo de sitio (*type*), el nombre (*place_name*), la calificación (*rate*) y el número de comentarios (*num_comments*).

lat	lng	url	type	place_name	rate	num_comments
-0.221435	-78.51109	https://www.tripadvisor....	restaurant	La Purisima	4.5	321
-0.221384	-78.51551	https://www.tripadvisor....	restaurant	Casa Gangotena	4.5	543
-0.220568	-78.51213	https://www.tripadvisor....	restaurant	Fabiolita	4.5	230
-0.221140	-78.50737	https://www.tripadvisor....	restaurant	Bandido Brewing	4.5	497
-0.221379	-78.51100	https://www.tripadvisor....	restaurant	Cafe Galletti	4.5	125
-0.225044	-78.51361	https://www.tripadvisor....	restaurant	Casa Los Geranios	4.0	209
-0.220296	-78.51498	https://www.tripadvisor....	restaurant	Tianguez	4.0	236
-0.230082	-78.51762	https://www.tripadvisor....	restaurant	Pim's Panecillo	4.0	388
-0.220508	-78.51214	https://www.tripadvisor....	restaurant	Dulceria Colonial	4.5	78

Tabla 3. Extracto de tabla de sitios turísticos de Quito con calificaciones y número de reseñas

2.2.5 El Comercio

La adquisición de noticias del sitio web de El Comercio se realizó también mediante *web scraping*. Este sitio proporciona un método de búsqueda de artículos basado en etiquetas. Como filtrado de datos se utilizó la búsqueda del patrón “La Floresta”, de acuerdo con la Ilustración 14.

Ilustración 14. Búsqueda por etiqueta “La Floresta” en sitio web de El Comercio (El Comercio, 2018)

El proceso de recolección de noticias empieza con identificar la forma en que El Comercio realiza la búsqueda por etiqueta. El patrón para buscar se inserta en una URL, a la que se accede automáticamente y se muestran los artículos encontrados. En la misma página, el sitio web despliega hasta veinte artículos encontrados, si existen más, el sitio web los divide entre varias páginas. Al final de cada página se encuentra una sección de navegación entre páginas, como se ve en la Ilustración 15, que se puede utilizar para obtener todas las páginas de resultados, así como los enlaces para poder acceder a las noticias.

Ilustración 15. Sección de navegación entre páginas de resultados (El Comercio, 2018)

Mediante una función de *web scraping* que extrae el contenido del elemento, de clase *pagination*, se pueden obtener un conjunto de URLs pertenecientes a las páginas de resultados. Accediendo a cada URL del conjunto, se encuentran más URLs correspondientes a las noticias. Esta estructura se puede representar mediante un árbol de profundidad igual a 2, mostrado en la Ilustración 16.

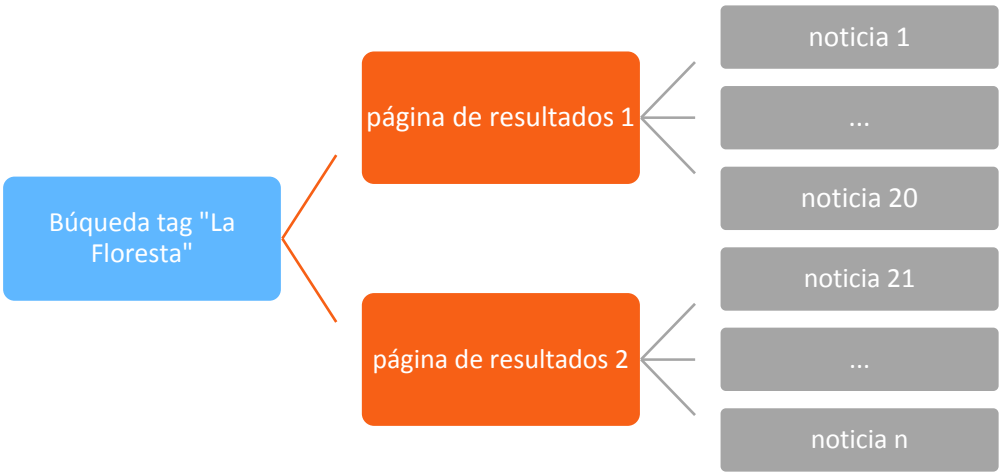


Ilustración 16. Estructura de acceso a noticias en sitio web El Comercio

Para recoger los links de las noticias, se accede a cada página y se extrae el elemento html que contiene la referencia (*href*), como aparece en la Ilustración 17.

Ilustración 17. Elemento de página que contiene enlace a una noticia en El Comercio (El Comercio, 2018)

Así, se pueden programar las funciones para recoger las URLs hacia cada noticia. El siguiente paso es acceder a cada URL de noticias y extraer los datos deseados, en este caso, la fecha, el título de la noticia y el texto. Después de examinar la estructura de las páginas de noticias, se determinó que el título se encuentra en el elemento html de clase *title*, el texto en elementos de clase *paragraph* y la fecha en el elemento *publishDate*. La implementación de la función de *web scraping* que realiza este proceso de obtención de noticias es la que se muestra en el Código 6.

```
# Archivo: get_ec.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Obtener título
title <- webartpage %>%
  html_nodes('[class="title"]') %>%
  html_nodes('h1') %>%
  html_text()

# Obtener texto
parrafos <- webartpage %>%
  html_nodes('[class="paragraphs"]') %>%
  html_nodes('p') %>%
  html_text()

# Obtener fecha
```

```
fecha <- webpage %>%
  html_nodes('[class="article"]') %>%
  html_nodes('[class="two-cols-article"]') %>%
  html_nodes('[class="left-col"]') %>%
  html_nodes('[class="publishDate"]') %>%
  html_text()
```

Código 6. Funciones de web scraping para obtener datos de noticias de El Comercio

Los datos adquiridos de las páginas de noticias, que se encuentran en todas las páginas de resultados, se almacenaron en una tabla de cuatro campos donde se almacena la fecha (*date*), la URL (*link*), el título de la noticia (*title*) y el texto (*text*). Un extracto de esta tabla se muestra en la Tabla 4.

date	link	title	text
2018-05-06	http://www.elcom...	La tercera edici...	El sábado 5 y el domingo 6 de mayo, las calles de La Flor...
2018-04-17	http://www.elcom...	Trabajos de bac...	La Empresa Pública Metropolitana de Obras Públicas del...
2018-03-14	http://www.elcom...	Estudiantes de ...	Siete estudiantes de fotografía fueron asaltados la noch...
2018-02-07	http://www.elcom...	Tráfico lento po...	Quito amaneció nublado la mañana de este miércoles, 7...
2018-01-22	http://www.elcom...	Un accidente d...	Un automóvil Skoda rojo se accidentó la mañana de est...
2017-12-09	http://www.elcom...	Un vehículo ap...	Un vehículo terminó en el interior del redondel de La Fl...
2017-11-10	http://www.elcom...	La planificación...	En Quito hay seis barrios que tienen ordenanzas propia...
2017-11-01	http://www.elcom...	La ordenanza q...	La Floresta cuenta con una Ordenanza especial para su ...
2017-09-26	http://www.elcom...	El parque de N...	El parque Navarro está de aniversario. Tras su renovació...
2017-09-03	http://www.elcom...	La Floresta luch...	La Floresta del siglo XXI es bipolar. Es un barrio que nav...

Tabla 4. Extracto de tabla de noticias sobre La Floresta de El Comercio

2.2.6 El Telégrafo

El sitio web de El Telégrafo funciona de similar manera al de El Comercio. Proporciona un buscador de noticias, como el indicado en la Ilustración 18, con una ventaja adicional para el *web scraper*. El buscador de este sitio web puede mostrar todos los resultados en una misma página.

Ilustración 18. Búsqueda por etiqueta "La Floresta" en sitio web de El Telégrafo (El Telégrafo, 2018).

La estructura de acceso hacia las noticias de El Telégrafo, puede representarse mediante un árbol de profundidad igual a 1, como se observa en la Ilustración 19, el cual es más simple que la estructura de acceso hacia las noticias de El Comercio, de estructura de acceso mostrada en la Ilustración 16.

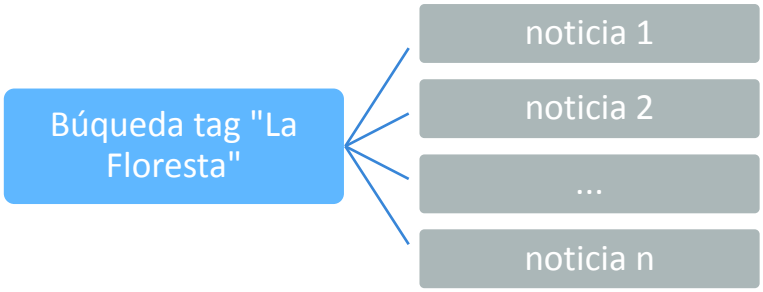


Ilustración 19. Estructura de acceso a noticias en sitio web El Telégrafo

Así, sólo se realiza una recolección de URLs de noticias. En la página de búsqueda, las URLs deseadas se encuentran en la referencia dentro del elemento de clase result-title, como se ve en la Ilustración 20.

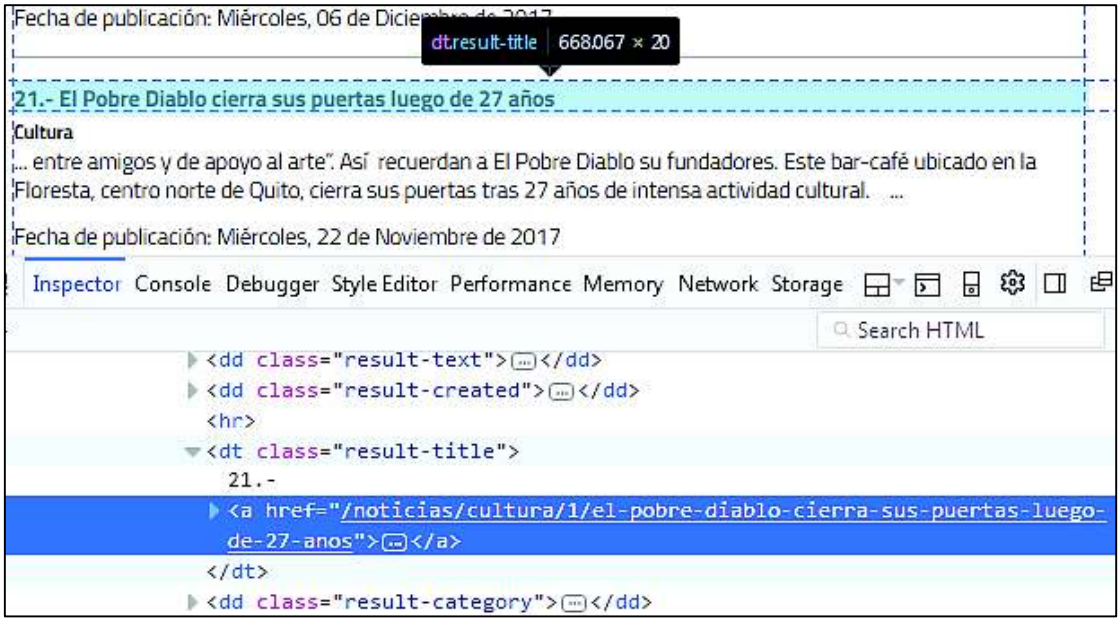


Ilustración 20. Elemento que contiene el enlace a una noticia en página de El Telégrafo (El Telégrafo, 2018)

Una vez recogidas las URLs, se puede usar un bucle para acceder a cada una y recoger los datos de las noticias. En la página de cada noticia de El Telégrafo, el título se encuentra en el elemento html de clase *story-header-block.h1*, la fecha en el elemento de clase *story-publishup* y el texto en el elemento *articleBody*. Las recolección de estos elementos se realiza mediante las funciones del Código 7.

```
# Archivo: get_et.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Obtener título
title <- webpage %>%
  html_nodes('[class="story-header-block"]') %>%
  html_nodes('h1') %>%
  html_text()

# Obtener fecha
date <- webpage %>%
  html_nodes('[class="story-publishup"]') %>%
  html_text()

# Obtener texto
text <- webpage %>%
  html_nodes('[itemprop="articleBody"]') %>%
  html_nodes('p') %>%
  html_text()
```

Código 7. Funciones de web scraping de noticias de El Telégrafo

Después de capturadas, las noticias acerca de La Floresta del diario El Telégrafo, se almacenan en una tabla de campos: fecha (*date*), URL (*link*), título de la noticia (*title*) y texto (*text*). Tal como se muestra en la Tabla 5.

date	link	title	text
2017-05-06	http://www.elteleg...	La Floresta cumple 100 años con ...	Hace 100 años se lotizó La Floresta, una hacienda de la f...
2017-03-20	http://www.elteleg...	Deslizamiento de tierra produjo ...	Los organismos de emergencia reportaron esta mañana ...
2016-07-28	http://www.elteleg...	Dos apuñalados en sector de co...	"Hace un mes el Alcalde dijo que este era el barrio más s...
2015-12-17	http://www.elteleg...	La cultura autogestionada está e...	Hace poco más de un siglo que existe el barrio la Florest...
2014-04-13	http://www.elteleg...	La Floresta, un barrio que guard...	En la zona, que pertenece a la parroquia itchimbía, viven...
2012-11-02	http://www.elteleg...	Un hombre es baleado en instan...	De dos impactos de bala fue asesinado un hombre en el...
2018-05-25	http://www.elteleg...	No Lugar se muda a La Tola y pre...	Por su octavo aniversario este espacio expositivo se cam...
2018-05-15	http://www.elteleg...	"La sociedad de lavanderas", un r...	El documental es codirigido por la cineasta Lynne Sachs ...
2018-05-03	http://www.elteleg...	Entre memorias y olvidos, David ...	El fotógrafo documental ganó la última edición del Pre...
2018-04-20	http://www.elteleg...	La González Suárez es la tercera z...	En la avenida González Suárez, ubicada al norte de Quit...
2018-04-01	http://www.elteleg...	Un poeta cuya vida fue marcada ...	El estrés por el sismo de 1949, que nunca olvidó, le gene...
2018-03-06	http://www.elteleg...	Crear un ecosistema de innovaci...	La naturaleza multidisciplinaria de la comunidad de emp...

Tabla 5. Extracto de tabla de noticias sobre La Floresta de El Telégrafo

Luego de obtenerse los datos, un proceso de conversión de formatos y estandarización de campos es necesario. Este proceso se lleva a cabo en la siguiente fase 2.3.Extracción de datos.

2.3 Extracción de datos

La fase de extracción tiene el objetivo de transformar datos de distintos formatos en uno que la solución de analítica pueda utilizar (Erl, Khattak, & Buhler, 2016). Por lo general, los datos provenientes de una misma fuente pueden parecerse entre sí en su forma, pero difieren con las de otras fuentes. Esto hace necesario cierto nivel de estandarización en la identificación de los atributos de los datos (Ganis & Kohirkar, 2016). La estandarización asegura que todos los datos de un mismo campo sean forzados a ser parte de un estándar, facilitando su comprensión y ofreciendo una mejor estética. Para la realizar la Agregación y representación de datos, es necesario que las columnas posean estándares, de no ser así, las operaciones entre ellos resultarán en valores erróneos (Loshin, 2002).

En esta fase, se realizaron fundamentalmente dos tareas, la conversión de formatos de datos y la estandarización de campos. La primera, Conversión de formatos, se realizó inmediatamente después de recibir los datos filtrados, esto fue necesario para su inmediata manipulación. Y la segunda, Estandarización de campos, se realizó de manera conjunta con la fase de Validación y Limpieza de datos Validación y limpieza de datos. A continuación, se detalla la forma en que se realizaron ambas tareas:

2.3.1 Conversión de formatos

Al utilizar Python, para obtener posts de la API oficial de Facebook, el programa recibe un objeto JSON. Python no procesa objetos JSON directamente, sino que los transforma a una estructura de datos parecida en sintaxis, a estos datos se los llama diccionarios. Una vez que se tienen los diccionarios en memoria, el programa carga los posts a la base de datos no relacional en CouchDB, usando la librería oficial *couchdb* para Python. En Twitter, por otro lado, al usar la librería *GetOldTweets*, se obtienen los atributos de cada tweet por separado. A partir de esto, se construyen diccionarios de *tweets que* son subidos a la base de datos.

Para la conversión de datos de las demás fuentes, en las que se utilizó R, la estructura de datos utilizada fue el *dataframe*. Durante la adquisición, los programas tuvieron que recibir datos en formatos JSON y HTML. JSON es fácilmente transformado a listas de R, las cuales pueden manipularse de forma similar a los *dataframes*. Por otro lado, los datos en HTML requieren de un proceso de interpretación. Esta interpretación de páginas HTML se

puede realizar con las librerías *rvest* y *xml*, con las que se puede acceder y extraer nodos específicos de la página.

2.3.2 Estandarización de campos

Es común que, al adquirir datos de diversas fuentes, éstos se encuentren con atributos similares nombrados de forma diferente. Por ejemplo, al obtener un post de Facebook, el campo fecha se denomina *created*_time, mientras que en Twitter se lo recibe como *date*, como se muestra en la Ilustración 21. Otro campo en el que difieren es en el texto de la publicación; en Facebook se denomina *message*, mientras que Twitter lo nombra *text*. Y mientras en Facebook se llaman *likes*, en Twitter se denomina *favorite_count*. Todo este tipo de inconsistencias fueron mitigadas mediante una estandarización en los nombres de los campos. Además, las fechas fueron formateadas con la misma estructura año-mes-día.

Campos \ Fuentes	Facebook		Twitter	
	nombre de campo	formato	nombre de campo	formato
id	id		id	
date	created_time	AAAA-MM-DD T h:m:s + ####	date	MM/DD/AAAA h:m:s
text	message		text	
author	faltante		name	
link	faltante		source	
likes	likes/summary/total_count		favorite_count	
location	faltante		location	
hashtags	faltante		hashtags	
user_mentions	faltante		user_mentions	
retweet_count	faltante		retweet_count	

Estandarización de campos

Campos \ Fuentes	Facebook		Twitter	
	nombre de campo	formato	nombre de campo	formato
id	id		id	
date	date	AAAA-MM-DD	date	AAAA-MM-DD
text	text		text	
author	faltante		name	
link	faltante		source	
likes	likes		likes	
location	faltante		location	
hashtags	faltante		hashtags	
user_mentions	faltante		user_mentions	
retweet_count	faltante		retweet_count	

Ilustración 21. Ejemplo de estandarización de campos entre datos de Facebook y Twitter

La Ilustración 21 muestra el proceso de estandarización de campos se llevó a cabo para los datos de cada una de las fuentes, excepto para Trip Advisor, cuyos datos no entraron en el proceso de análisis de texto, sino que se extrajeron las ubicaciones geográficas, nombres, número de reseñas y tipo de los sitios de interés, para pasar a mostrarlos directamente sobre un mapa en la fase 2.7.Visualización de datos. Para conocer los cambios en los campos de las demás fuentes, ver el ANEXOS

Anexo 1. Registro de estandarización de campos.

2.4 Validación y limpieza de datos

La fase de validación y limpieza de datos se encarga de establecer reglas de validación y eliminación de datos inválidos conocidos, con el fin de aportar calidad al conjunto de datos. Los datos inválidos pueden distorsionar el análisis y hacer que éste arroje resultados erróneos (Erl, Khattak, & Buhler, 2016). En esta fase, se deben examinar la integridad y racionalidad de los datos; entre las tareas están la creación de rutinas para la eliminación de duplicados, el tratamiento de datos faltantes y la definición de restricciones que aseguran validez de los datos (Chen, Mao, & Liu, 2014).

En esta fase se configuraron reglas o criterios que se ejecutan automáticamente sobre los datos y que aseguran que éstos llegarán íntegros y coherentes al análisis. En los datos recogidos de todas las fuentes, se aplicaron los mismos criterios de validación y limpieza.

La limpieza se basa fundamentalmente en la eliminación de registros que no poseen texto para analizar. Mientras que los criterios de validación se definieron después de examinar de forma general de los datos obtenidos; la mayoría de los datos inválidos correspondían a artículos que no se referían al barrio La Floresta de Quito, sino al barrio del mismo nombre ubicado en Guayaquil, o a dos ciudades de Brasil de nombres “La Alta Floresta” y “Floresta”.

Por esta razón, los criterios de validación y limpieza definidos son:

- Si el campo texto está vacío, eliminar registro.
- Si el campo texto no posee la palabra “Quito”, pero sí una de las siguientes:
  - “Guayaquil” o “GYE”
  - “Brasil”
  - “Alta Floresta”

  Eliminar registro.

Estas tareas se realizaron mediante funciones básicas de R para manipulación de texto. Los datos sin preprocesar son leídos de la base de datos y son representados en *dataframes*, donde mediante la función *grep*, se identifica la posición de los elementos que cumplan con un patrón de texto, pasado como parámetro en la función. Este parámetro depende directamente de los criterios de limpieza y validación. Finalmente, se quitan los elementos identificados como no válidos o vacíos. Un ejemplo del uso de *grep*, se muestra en la Ilustración 22.

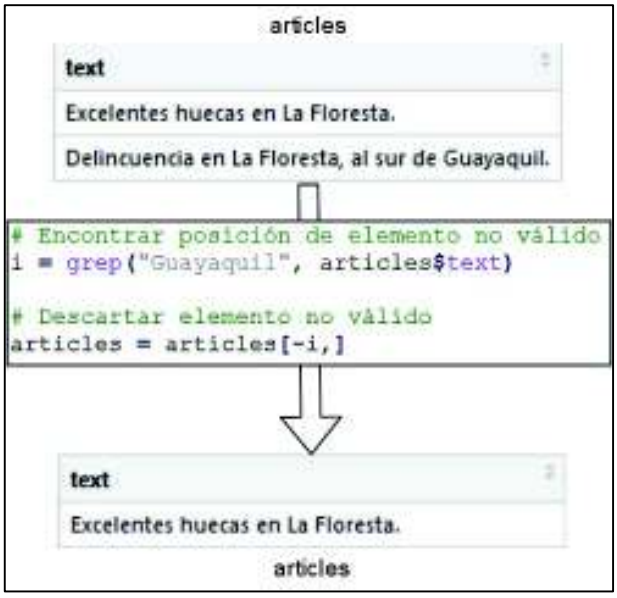


Ilustración 22. Ejemplo del uso de grep para limpieza de datos

En la Ilustración 22 se muestra el uso de *grep* para encontrar la posición *i* del elemento de la tabla *articles*, que contenga el patrón “Guayaquil” en el campo *text*. Una vez encontrado el elemento no válido de posición *i*, el último paso es quitarlo de la tabla *articles*.

Para el caso de los datos provenientes de The Culture Trip, tuvo que realizarse un paso adicional. Todo el contenido publicado en esta fuente es texto en inglés, por lo que fue necesario incorporar una función de traducción. Esta función se basa en el envío del texto a la API de *Google Translate*, y la respuesta traducida es almacenada en lugar del texto en inglés. La desventaja de Google Translate, en su versión gratuita, es que tiene un límite de traducción de hasta 5000 caracteres. La solución a este problema fue la división del documento en fracciones de menos de 5000 caracteres cada una. La función de traducción se puede ver en el Anexo 2. Función de traducción usando R y la API de Google Translate.

Después de realizar la validación y limpieza, los datos limpios y validados son almacenados en nuevas bases de datos. Un hecho observable, como se muestra en la Ilustración 23. b, es que la cantidad de documentos en las bases de datos iniciales es, para todos los casos, mayor a la cantidad de documentos en las bases de datos limpios. La forma en que se encuentran almacenados, tanto los datos sin preprocesar como los datos limpios, se detalla en la siguiente sección Agregación y representación de datos.

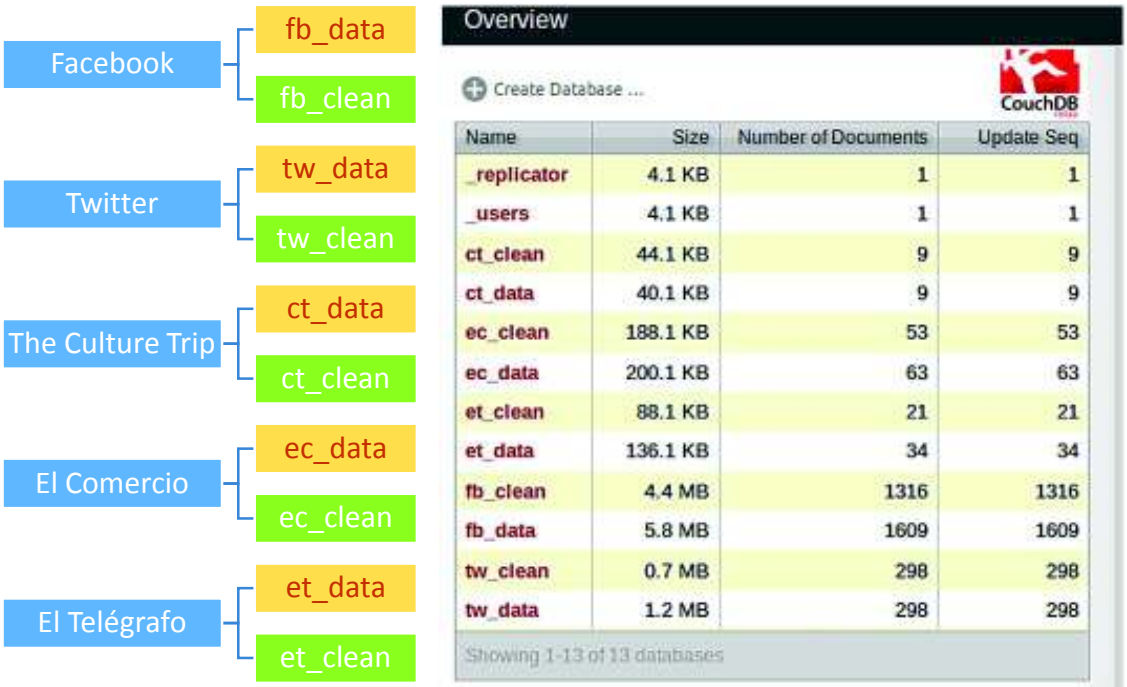
2.5 Agregación y representación de datos

Frecuentemente los datos relevantes no se encuentran en un solo conjunto de datos, sino que provienen de distintas fuentes y deben ser unificados en base a sus características (Runkler, 2016). La solución podría demandar el establecimiento de un repositorio central unificado, como por ejemplo, una base de datos no relacional o *Not-only SQL* (NoSQL). Una base de datos NoSQL es una base de datos no relacional de alta escalabilidad y tolerancia a fallos, con la función de almacenar específicamente datos semi-estructurados y no estructurados (Erl, Khattak, & Buhler, 2016).

El sistema de base de datos seleccionado para el proyecto fue CouchDB. La razón principal es, que este gestor de bases de datos utiliza documentos JSON como formato nativo de almacenamiento, lo que se adapta perfectamente con la solución de analítica propuesta. CouchDB trabaja con documentos sin esquema fijo, lo que hace posible el almacenamiento de registros que pueden o no compartir campos en común. Además, una característica de CouchDB frente a otros sistemas de bases de datos NoSQL, es su facilidad de uso, lo que permite al analista centrarse más en el análisis y menos en la configuración y uso de ambientes complejos. CouchDB, a partir de su versión 2.0, puede trabajar sobre sistemas distribuidos, en este aspecto, la fortaleza de CouchDB es la alta disponibilidad de datos y tolerancia a distribuir la carga de datos entre varios servidores (Anderson, Lehnardt, & Slater, 2010). Para el presente proyecto utilizó CouchDB 1.6; sin embargo, queda abierta la posibilidad de escalar el proyecto a sistemas distribuidos en el futuro, siendo necesaria la actualización de CouchDB a una versión más actual.

El flujo de datos en el sistema, como se puede observar en la arquitectura de la Ilustración 1, está en función de las fases de analítica. Básicamente, el almacenamiento de los datos se realiza en tres partes. Las dos primeras corresponden a la agregación de datos antes y después del preprocesamiento en un repositorio unificado. Y la tercera parte, se realiza después del análisis, donde los resultados en forma de resúmenes de datos se almacenan en archivos de texto plano para después ser leídos por la capa de visualización.

Al utilizarse una base de datos NoSQL, el modelo de datos no se basa en esquemas de tablas y relaciones. En su lugar, CouchDB maneja bases de datos completamente separadas, como se puede observar en la Ilustración 23.



a) *b)*

Ilustración 23. a) Modelo de bases de datos por fuente. b) Bases de datos en CouchDB

En la Ilustración 23, se muestra que para cada fuente fueron creadas dos bases de datos, una donde se almacenan los datos recogidos antes del preprocesamiento (color naranja), y otra para almacenar los datos preprocesados o limpios (color verde). En el caso de Facebook, por ejemplo, *fb_data* es la base de datos previos al preprocesamiento y *fb_clean* es la base de datos preprocesados.

Para el caso de los datos de *Trip Advisor*, los cuales son utilizados únicamente para graficar los sitios en un mapa y no para análisis de texto, se construyó un archivo de texto plano en formato *Comma-Separated Values* (CSV), que es leído directamente por la capa de visualización, sección 2.7.Visualización de datos.

2.6 Análisis de datos (Minería de texto)

El objetivo de la minería de texto es analizar y descubrir patrones de interés, incluyendo tendencias y anomalías en datos textuales (Zhai & Aggarwal, 2014). Los datos textuales pueden representarse como una cadena secuencial de palabras, o como una bolsa de palabras[3]. El enfoque de bolsa de palabras es el más utilizado en minería de texto. En este enfoque, las palabras se denominan términos, mismos que se encuentran agrupados en documentos; una colección de texto de *n* documentos y *d* términos corresponde a una matriz *n x d*, conocida como matriz documento-término. Una colección de documentos se denomina *corpus*. Una vez que el texto es representado como un vector en el espacio, los datos pueden considerarse datos cuantitativos multidimensionales, donde las palabras son atributos y como valores llevan sus frecuencias (Aggarwall, 2015).

Existen varios métodos de minería de datos, cada uno orientado hacia diversos propósitos, la minería de texto, al ser parte de minería de datos, utiliza adaptaciones de varias de sus técnicas convencionales de análisis (Aggarwall, 2015). Se pueden clasificar los métodos de minería de datos en dos grandes grupos: los orientados a verificación, que buscan confirmar una hipótesis; y los orientados a descubrimiento, los cuales buscan patrones de forma autónoma en los datos (Maimon & Rokach, 2010). En la Ilustración 24 se muestra la taxonomía o clasificación de las técnicas de minería de datos, aquellas que son exclusivamente de minería de texto se encuentran señaladas con *(tm, de text mining)*.

Se han desarrollado una gran cantidad de técnicas de análisis de datos orientadas a descubrimiento. Estas técnicas, se aplican en varios dominios de datos, tanto cuantitativos como cualitativos. Los datos textuales, al poder modelarse como datos cuantitativos a través de las frecuencias de las palabras, pueden ser tratados con la mayoría de los métodos orientados a datos cuantitativos (Zhai & Aggarwal, 2014).

[3] Bolsa de palabras, inglés *bag of words*, es una forma de representación de texto en la que se ignora el orden de las palabras (Aggarwall, 2015).

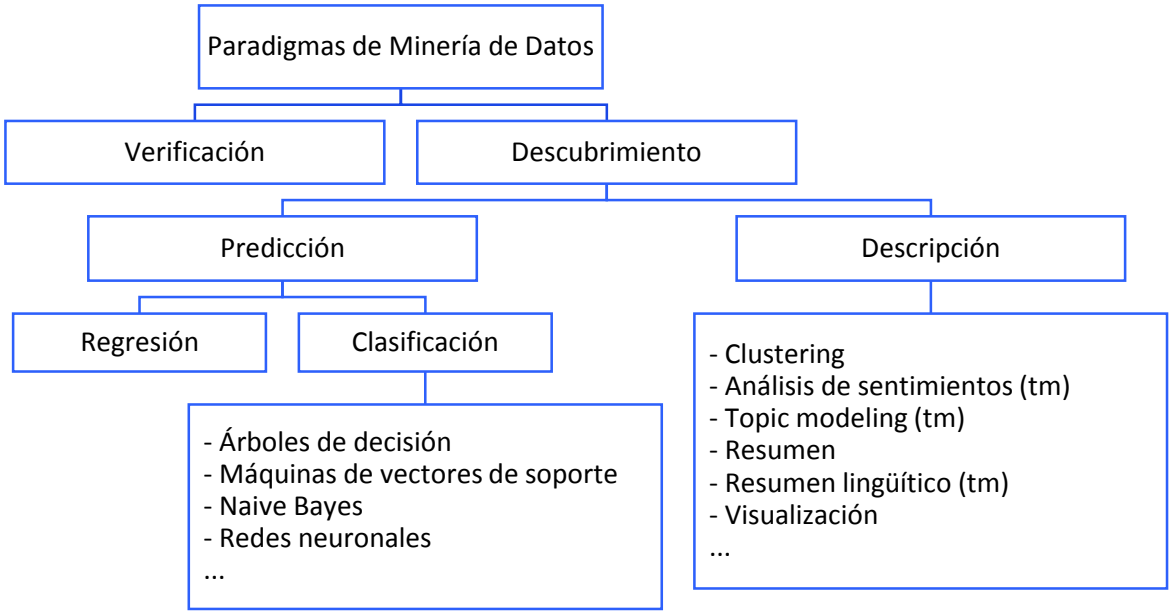


Ilustración 24. Taxonomía de minería de datos (Maimon & Rokach, 2010)

Para el desarrollo del presente proyecto, se utilizaron tanto técnicas de clasificación como de descripción de datos, esto se muestra en las secciones 2.6.1.Clasificación de datos y 2.6.2.Descripción de datos. La clasificación de datos se basa en la construcción y aplicación de un modelo predictivo, que utiliza lo que en aprendizaje de máquina[4] se conoce como aprendizaje supervisado, para la estimación de valores faltantes a partir de un conjunto de datos de entrenamiento de valores conocidos. Respecto a clasificación, se probaron algoritmos de aprendizaje de máquina como: los árboles de decisión, máquinas de vectores de soporte y Naïve Bayes, en busca del mejor clasificador. Este clasificador tiene la tarea de asignar etiquetas a los datos según la temática a la que pertenezcan. Por otro lado, como métodos descriptivos de datos, se utilizó: análisis de sentimientos, para determinar si los textos expresan sentimientos positivos o negativos, y para detectar emociones; *topic modeling*, que permite el descubrimiento de tópicos en el texto; y otros tipos de resúmenes de datos como las tablas de frecuencias, para contabilizar las publicaciones analizadas.

[4] Aprendizaje de máquina (en inglés *machine learning*) puede definirse como la capacidad de un sistema de cómputo para detectar patrones y realizar tareas de forma autónoma.

La fase de análisis, durante el desarrollo de la aplicación, se basó en analizar los datos históricos, que fueron recolectados hasta el 15 de octubre de 2017. Después, las técnicas utilizadas en el proceso fueron modificadas para funcionar de manera automática cuando el sistema sea puesto en recolección y análisis continuo de datos. El proceso de análisis fue el siguiente:

1. *Carga de datos a memoria*

El primer paso es la descarga de nuevos datos alojados en cada una de las bases de datos. En el caso del primer análisis realizado con datos históricos, todos los datos recolectados y limpios entraron en el proceso. En la descarga de datos a memoria, se utiliza la librería *RCurl* para solicitar datos desde R a CouchDB, a través de un método GET enviado a sus APIs. Una vez recibidos los documentos del conjunto de bases de datos limpios de todas las fuentes, se aplica una función que se creó para construir una versión unificada y simplificada de estos; dicha función guarda en un *dataframe* los datos junto a sus tres atributos de mayor importancia: el texto (*text*), la fuente (*source*) y la fecha de creación (*date*), tal como se observa en la Tabla 6.

date	source	text
2017-02-10	fb	Artesanía fina para regalar en el día de El Amor. Un poco...
2017-02-11	fb	Amaneció sábado y hoy nos vamos en Bici a la Feria! Est...
2017-02-11	fb	La floresta está llena de amor, diseño, buena comida y g...
2017-02-11	fb	Hoy y mañana tienes las mejores opciones y ofertas llen...
2017-02-12	fb	Hoy seguimos enamorándonos en La Floresta. Empezam...
2017-03-04	fb	No olvides que los Sábados la Floresta se pone de Fiest...
2017-03-04	fb	Música , mural en vivo, feria de pulgas y asadito de amig...
2017-07-25	fb	Si te gusta pedalear y compartir del arte, diseño, gastro...
2014-01-03	tw	Plan de Viernes por la noche! #friday #night #ps3 #gtav ...
2014-01-06	tw	#magia #pic #buenoDias @La Floresta http:// instagram...
2014-01-06	tw	::.."Soy lo que yo digo." Desde arriba...:: @La Floresta htt...
2014-01-07	tw	::..Con plata todos vuelan..:: @La Floresta http:// instagr...
2014-01-07	tw	Récord???!!! Ja récord el viaje en bus desde la Floresta a ...
2014-01-08	tw	::..Pasaba por aquí y miren! Su rostro y su hogar. Lind@...
2014-01-11	tw	Iglesia La Floresta en peligro apoyanos manana en la rec...

Tabla 6. Extracto de tabla unificada de datos históricos

2. Limpieza de texto

Además de la fase de Validación y limpieza de datos, en donde se quitaron registros vacíos y no válidos, puede requerirse una limpieza adicional del texto, dependiendo del tipo de análisis a realizar. El enfoque de minería de texto utilizado en este proyecto es el de bolsa de palabras. Por lo que, caracteres como los signos de puntuación y demás caracteres especiales no son relevantes para el análisis.

Las librerías de R para minería de texto a utilizar son *tm* y *RTextTools*. *tm* se encarga de la limpieza de texto y la creación de matrices de frecuencias de palabras (CRAN Project, 2017). Y *RTextTools* se encarga de la clasificación y la aplicación de modelos de aprendizaje supervisado sobre los datos de texto (CRAN Project, 2015). La librería *tm* soporta idioma español, pero *RTextTools* no. Esto no afecta el proceso de análisis en nuestro caso, ya que *RTextTools* tiene el trabajo de leer etiquetas en los datos de entrenamiento y generar un modelo a partir de las frecuencias de las palabras; no se realiza un análisis de su significado, por lo que el idioma en que se encuentren no tiene relevancia. Sin embargo, la codificación de los caracteres propios del español, como la "ñ" o las vocales tildadas, provocan errores en la ejecución de las funciones de *RTextTools*, por lo que es necesario modificarlas. Por esta razón se creó la función mostrada en el Código 8, ésta cambia todas las vocales tildadas por vocales sin tilde y todas las "ñ" por "ni". Dicha operación se realiza solamente para el etiquetado de los textos; una vez finalizado, se utilizan los textos originales. Esto es necesario debido a que posteriormente, en Análisis de sentimientos, se hace uso de un diccionario en el cual, sí se hace uso del significado de cada término.

```
# Archivo: topic_SVM.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

cleanText <- function(texts_vect){

  # Definir diccionario de conversión
  dict  <- c('a','e','i','o','u','ü','A','E','I','O','U','ni','NI')
  names(dict) <- c('á','é','í','ó','ú','Ü','Á','É','Í','Ó','Ú','ñ','Ñ')

  # Quitar tildes y ñ
  for( n in 1:length(dict)){
    texts_vect <- gsub(names(dict)[n],dict[[n]],texts_vect)
    n=n+1
  }

  # Quitar URLs
  texts_vect <- gsub("(f|ht)tp(s?)://(.*)[.][a-z]+", "", texts_vect)
```

```
  texts_vect <- gsub("www+", '', texts_vect, perl=TRUE)

  # Eliminar caracteres extraños
  Encoding(texts_vect) <- "UTF-8"
  texts_vect <- iconv(texts_vect, "UTF-8", "ASCII",sub='')

  return(texts_vect)
}
```

Código 8. Función personalizada de limpieza y conversión de caracteres

La función recibe como parámetro un arreglo de textos. Además de quitar tildes y las "ñ", se quitan URLs alojadas en el texto, y cualquier carácter extraño o que no pueda ser procesado por *RTextTools*. Un ejemplo del resultado de esta función se puede observar en la Ilustración 25.

a) *b)*

Ilustración 25. a) Texto antes de la limpieza b) Texto después de la limpieza

En el caso de la aplicación de métodos de clasificación, se requiere construir la matriz de frecuencias de palabras. Para esto, *tm* requiere realizar una limpieza y transformación adicional del texto; se remueven los números, signos de puntuación, espacios en blanco y conjunciones, también se convierten todas las letras en minúsculas. El uso de *tm* para realización de estos pasos se muestra en el Código 9.

```
# Archivo: topic_SVM.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Crear el corpus
source <- VectorSource(textArray)
corpus <- Corpus(source)

# Remover números, puntuaciones, espacios y conjunciones
```

```
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeWords, stopwords('spanish'))

# Conversión a minúsculas
corpus <- tm_map(corpus, content_transformer(tolower))
```

Código 9. Pasos estándar de limpieza de texto

Como se puede observar, se necesita crear un objeto de tipo *corpus*, que contiene todos los documentos, para poder usar *tm*. Cabe señalar que, de todas las funciones aplicadas, la función *stopWords* es la única que recibe como parámetro el idioma del texto, esto es, para reconocer las conjunciones correspondientes a dicho idioma y removerlas.

Después de estos pasos, se pueden aplicar diversas técnicas de clasificación y descripción de datos. En las siguientes secciones, 2.6.1.Clasificación de datos y 2.6.2.Descripción de datos, se muestran las técnicas utilizadas para el desarrollo del sistema de analítica.

2.6.1 Clasificación de datos

La clasificación de datos, orientada a la minería de texto, se refiere a la clasificación de documentos de texto según categorías. Por lo general, no se busca determinar la categoría de un solo documento, sino de un conjunto de varios documentos no etiquetados o etiquetados de forma parcial, sobre los que se quiere inferir valores para sus etiquetas faltantes (Srivastava & Mehran, 2009). Para lograr el aprendizaje de un patrón, el aprendiz observa cada instancia t del conjunto de datos de entrenamiento, asociada al concepto *c(t)*. El trabajo del mecanismo de aprendizaje es estimar la función *c*, de tal manera que sea una generalización para todas las instancias del conjunto de entrenamiento y sea aplicable sobre instancias desconocidas. El clasificador observa las instancias del conjunto de entrenamiento y crea para cada una, una hipótesis específica. Entre las hipótesis creadas, deberán existir unas más generales que otras; es decir, hipótesis que son capaces de estimar no sólo a una instancia, sino a un conjunto de ellas. Gracias al principio de aprendizaje inductivo, se conoce que si una hipótesis puede estimar un patrón para un número lo suficientemente grande de instancias, ésta también será capaz de estimar para un conjunto de instancias desconocidas (Ashish & Avinash, 2016).

La clasificación, al tratarse de aprendizaje supervisado, requiere de un conjunto de datos de entrenamiento. De los 1721 datos históricos recolectados hasta el 15 de octubre de

2017, se seleccionó aleatoriamente el 20% de ellos para cumplir el rol de datos de entrenamiento; 344 datos. El propósito de usar clasificación en el sistema, es el de categorizar cada documento de entre las temáticas que más se hablan acerca del barrio La Floresta.

El proceso de selección y asignación manual de etiquetas se realizó iterativamente mediante prueba y error. El ejemplo se muestra en la Tabla 7:

id	text	label 1	label 2	label 3
1	Los artistas de Arupo Festival de Graffiti Caida y Limpia se toman todos los espacios de nuestro barrio.	Graffiti	Artes gráficas	Arte
2	En LA FLORESTA: Vinicio Vallejos Villota te invita a "TALLERES CONTINUOS DE ARTE" Lugar: 419 ART STUDIO Invitamos cordialmente a los talleres continuos de: Dibujo conceptual, Comic, Animacion, Retrato, Caricatura, Figura Humana y pintura.	Pintura	Artes gráficas	Arte
3	Talleres vacacionales de pintura, baile, musica, ceramica y paseos. Del 11 de julio al 5 de agosto.	Arte y paseos	Arte y paseos	Arte
4	Algunas razones por las que nos deberia preocupar mucho el notable aumento del trafico en las calles del barrio, y por las que deberiamos seguir trabajando hacia los objetivos contemplados en el plan especial del barrio: la progresiva pacificacion y peatonizacion de calles.	Tráfico	Movilidad	
5	Ven en Bici a las rutas controladas De La Floresta sella tu pasaporte en cada uno de los puntos de interes de nuestro mapa de la Feria de Verano 2017	Bicicleta	Movilidad	

Tabla 7. Ejemplo de asignación de etiquetas a datos de entrenamiento

La Tabla 7 muestra un ejemplo de cómo se seleccionaron y asignaron las etiquetas. La tabla consta de un identificador del documento (*id*), el texto (*text*), y las etiquetas (*label #*). Donde el número de etiqueta representa el número de iteración. El ejemplo es una abstracción simplificada del etiquetado manual realizado.

El proceso se basó en ir leyendo uno a uno cada documento e ir asignando temáticas según lo que trate cada texto (*label 1*), cuando se encuentran textos de temáticas similares como los documentos 1 y 2, y por otro lado el 4 y el 5, se busca una temática más general que represente todo el grupo. Esto se repitió hasta que todos los documentos queden

etiquetados con una categoría lo más general posible. Al final se determinó que todos los documentos del conjunto de entrenamiento podían clasificarse según siete categorías.

Las categorías a clasificar los datos y sus respectivos ejemplos, extraídos del conjunto de datos, se muestran a continuación:

- **Gastronomía**: textos con temáticas referentes a alimentación. Por ejemplo:

  "Por lo menos una vez a la semana, después del trabajo y antes de ir a la casa, con los compañeros buscamos huecas de comida típica. Desde hace muchos años no íbamos a La Floresta, al famoso Parque de las Tripas."

  Fuente: Twitter.

- **Arte**: textos con temáticas referentes a música, pintura, baile, teatro o cultura. Por ejemplo:

  "Hace unos 20 años, debido a su naturaleza atractiva y de bajo costo de la zona, una creciente comunidad de artistas y escritores comenzó a establecerse aquí, y la mezcla resultante de hogares clásicos y galerías de arte de vanguardia, así como la grandes murales pintados a los lados de los edificios, realzaron la reputación de La Floresta."

  (Traducido) Fuente: The Culture Trip.

- **Ambiente**: textos con temáticas referentes al cuidado del medio ambiente. Por ejemplo:

  "Vecinos, si tienen desechos especiales que no saben dónde botar como pilas, baterías, focos, lacas y pinturas, medicinas caducadas, aparatos eléctricos o electrónicos. Emaseo 24 horas contigo ha colocado un punto móvil de recolección en el parqueadero del Supermaxi 12 de Octubre."

  Fuente: Facebook.

- **Organización** barrial: textos con temáticas referentes a la comunidad y el barrio. Por ejemplo:

"Iglesia La Floresta en peligro apóyanos mañana en la recolección de firmas para salvarla, a las 9h en el Parque del Sector"

Fuente: Twitter.

- **Movilidad**: textos con temáticas referentes a la movilidad peatonal, de vehículos motorizados y no motorizados. Por ejemplo:

  "Un nuevo contraflujo se implementó desde esta semana en el sector de La Floresta, al norte de Quito. En la avenida Toledo, desde la avenida Ladrón de Guevara hasta la Madrid, tres de los cuatro carriles permiten que los vehículos circulen en sentido sur-norte."

  Fuente: El Comercio.

- **Inseguridad**: textos con temáticas referentes a la inseguridad en el barrio. Por ejemplo:

  "…Queridos vecinos de La Floresta... Este fin de semana fui víctima de un robo a mano armada, a horas de la noche cerca del Incine de la Lugo. Quiero que por favor se comunique por este medio que este tipo de incidentes están sucediendo en el barrio, para prevenir a la gente y por qué no, para unirnos de alguna manera y cuidarnos los unos a los otros."

  Fuente: Facebook.

- **Otro**: textos que no encajan en ninguna de las anteriores categorías. Por ejemplo:

  "Receso de la oficina en el @Parque La Floresta…"

  Fuente: Twitter.

Una vez asignadas las etiquetas al conjunto de datos de entrenamiento, se pueden probar varios modelos de aprendizaje supervisado, para comparar sus exactitudes y seleccionar el mejor. Un extracto de los datos etiquetados se muestra en la Tabla 8. Donde en el campo *source*, los códigos fb, tw y ec corresponden a las fuentes Facebook, Twitter y El Comercio, respectivamente.

date	source	topic	text
2017-10-25	fb	Gastronomia	Sabias que tenemos #cafetería?? Tenemo...
2016-06-21	tw	Arte	Atardecer en la Floresta . Full inspiración!...
2016-11-09	fb	Arte	Quieres dictar tu Taller o curso en Arte Ac...
2017-05-07	fb	Arte	La tarde en #LaMADREdelasFerias vivela e...
2017-10-16	fb	Gastronomia	Estos son nuestros afortunados ganador...
2016-07-22	tw	Ambiente	La Floresta #arupo #pink #tree #color #l...
2016-08-25	fb	Arte	¡Este sábado! Recorre los Talleres Abierto...
2015-05-18	ec	Inseguridad	Los armados ingresaron al bar-tienda de ...

Tabla 8. Extracto del conjunto de datos de entrenamiento

Después de etiquetar el conjunto de datos de entrenamiento, aplicamos las funciones de Código 8 y de Código 9. Hecho esto, se tendrá un corpus preparado, con el cual se crean las siguientes matrices de frecuencias de palabras: matriz documento - término[5] (*DocumentTermMatrix*) y la matriz frecuencia de término - frecuencia inversa de documento[6] (*weightTfIdf*), mediante el Código 10.

```
# Archivo: topic_SVM.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Crear matrices de frecuencias de palabras
mat <- DocumentTermMatrix(corpus)
mat4 <- weightTfIdf(mat)
mat4 <- as.matrix(mat4)
```

Código 10. Creación de matrices de frecuencias de palabras

A continuación, se muestran los resultados de probar tres de las técnicas más utilizadas en aprendizaje de máquina. Todas mediante funciones de una librería llamada *e1071*, que es una librería de R para generar modelos de clasificación, realizar clustering, cálculo de

[5] Matriz documento – término es una matriz de frecuencias, donde se asocia el número de apariciones de las palabras con cada documento.

[6] Matriz frecuencia de término – frecuencia inversa de documento asocia el número de apariciones de las palabras con la colección completa de documentos o corpus (Aggarwall, 2015).

la ruta más corta, entre otras aplicaciones. Al final, el modelo de mayor exactitud se utilizará para implementarlo dentro del sistema de minería de texto.

2.6.1.1 Naïve Bayes

Un clasificador Naïve Bayes, es un clasificador probabilístico, que cuantifica mediante probabilidad, la relación entre los atributos y las clases. Fundamentalmente, este método hace uso del teorema de Bayes, cuya ecuación se muestra en la Ecuación 1, el cual se basa en conocimientos previos de los atributos para generar una distribución de probabilidades de correspondencia con cada clase. De esta manera, para etiquetar instancias desconocidas, se observan sus atributos y se determinan las probabilidades de correspondencia hacia cada clase; la clase con la probabilidad más alta es la que será asignada como etiqueta. Este tipo de clasificador funciona bajo dos supuestos: que los atributos del conjunto de datos son independientes entre ellos y que todos los atributos tienen la misma importancia (Ashish & Avinash, 2016).

$$P(A/B) = \frac{P(B/A) * P(A)}{P(B)}$$

Ecuación 1. Teorema de Bayes

Naïve Bayes genera un modelo de probabilidades a partir de los textos de entrenamiento, en base a las frecuencias de las palabras. Si un dato sin etiquetar, posee frecuencias de palabras similares al de un dato de etiqueta conocida, es probable que ambos pertenezcan a una misma categoría.

Para el presente proyecto, se utilizó Naïve Bayes para el etiquetado automático de documentos, en donde se desea averiguar a qué temáticas pertenecen a partir del conjunto de datos de entrenamiento etiquetados manualmente. Según Kloo (2015), una buena regla general para la selección del porcentaje de datos de entrenamiento y el porcentaje de datos de prueba, es el 80% y 20% respectivamente. Por lo tanto, para cada modelo, se utilizarán 275 registros para entrenar el modelo y 69 para ponerlo a prueba.

```
# Archivo: topic_bayes.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Inicio ejecución
t_ejec <- Sys.time()

# Crear clasificador con los datos de entrenamiento
```

```
classifier <- naiveBayes(mat4[1:275,], x$topic[1:275])

# Predecir etiquetas para los datos de prueba
predicted <- predict(classifier, mat4[276:344,])

# Calcular exactitud
bayes_accuracy <- recall_accuracy(as.character(x$topic[276:344]),
   as.character(predicted))

# Fin ejecución
t_ejec <- Sys.time() - t_ejec

# Mostrar exactitud y tiempo de ejecución
print(paste('bayes_accuracy =', round(bayes_accuracy,4)))
print(paste('tiempo_ejec =', round(t_ejec,2)))
```

Código 11. Cálculo de exactitud y tiempo de ejecución del clasificador basado en Naïve Bayes

En el Código 11 se muestra la generación de un clasificador basado en Naïve Bayes. Donde *mat4* es la matriz documento-término, *x* es el *dataframe* de datos de entrenamiento, las etiquetas se encuentran en la columna *topic*. Los documentos del 1 al 275 corresponden al conjunto de entrenamiento, mientras los datos del 276 al 344 corresponden al conjunto de prueba. La exactitud (*accuracy*) se calcula dividiendo el número de etiquetas estimadas correctamente del conjunto de datos de prueba, para el número de datos de prueba. El resultado de ejecutar el programa se muestra en Ilustración 26.

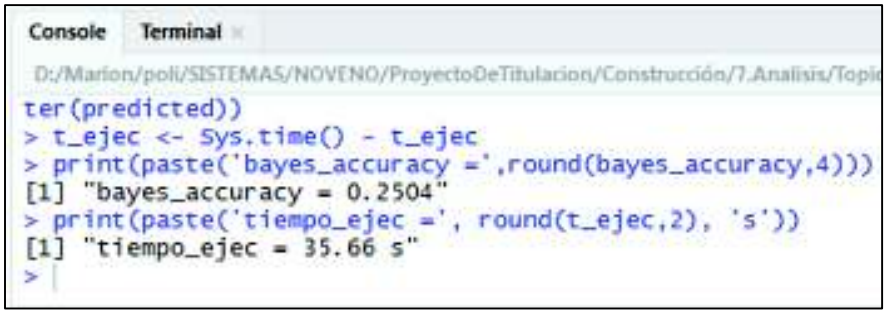


Ilustración 26. Exactitud y tiempo de ejecución del clasificador basado en Naïve Bayes

Este resultado quiere decir que aproximadamente el 25.04% de las etiquetas del conjunto de datos de prueba, fueron estimadas de forma correcta. El tiempo estimado de ejecución fue de 35.66 segundos.

2.6.1.2 Árboles de decisión

Los árboles de decisión, son clasificadores que se basan en la división recursiva del conjunto de datos de entrenamiento, al cual se le aplica una serie de condiciones sobre los

valores de los atributos del texto. Las condiciones, en clasificación de texto, corresponden normalmente a la evaluación de presencia o ausencia de una o más palabras en el documento (Ashish & Avinash, 2016).

Un árbol de decisión, es un árbol direccionado compuesto por: un nodo raíz, que es un nodo único que no posee flechas entrantes; nodos internos, conocidos como nodos de prueba; y nodos hoja, que son aquellos nodos terminales o de decisión. Cada nodo interno del árbol de decisión divide el espacio de instancias en dos o más sub-espacios, en base a una función discreta de evaluación aplicada sobre los valores de los atributos de entrada. En el caso de atributos numéricos, como las frecuencias de palabras, la condición corresponde a un rango. Por otro lado, los nodos hoja corresponden a las categorías en las que se quiere dividir el espacio de datos. Cada instancia, para ser categorizada, recorre desde la raíz, pasando por los nodos de evaluación, hasta llegar a los nodos hoja. El objetivo es dividir los datos en regiones tan puras como sean posibles; es decir, en grupos donde se tengan la mayor cantidad posible de instancias de la misma clase (Maimon & Rokach, 2010).

Para este caso, se utilizaron los mismos datos de prueba y entrenamiento que en Naïve Bayes, el cálculo de la exactitud del modelo generado mediante árboles de decisión se realiza como se muestra en el Código 12.

```
# Archivo: topic_tree.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Inicio ejecución
t_ejec <- Sys.time()

# Crear contenedor del conjunto de datos de prueba y entrenamiento
container <- create_container(mat, x$topic,
   trainSize=1:275,testSize=276:344, virgin=FALSE)

# Crear clasificador definiendo el tipo y el kernel
model <- train_model(container, 'TREE')

# Aplicar el modelo
results <- classify_model(container, model)

# Calcular exactitud
tree_accuracy <- recall_accuracy(x$topic[276:344],
   results[,"TREE_LABEL"])

# Fin ejecución
t_ejec <- Sys.time() - t_ejec

# Mostrar exactitud y tiempo de ejecución
```

```
print(paste('tree_accuracy =',round(tree_accuracy,4)))
print(paste('tiempo_ejec =', round(t_ejec,2), 's'))
```

Código 12. Cálculo de exactitud y tiempo de ejecución del clasificador basado en árboles de decisión

Para la creación de un modelo de clasificación basado en árboles de decisión, como se observa en Código 12, se crea un contenedor (*container*), con la matriz documento-término, el arreglo donde se encuentran las etiquetas (*x$topic*), se especifican cuáles de los datos son los de entrenamiento (*trainSize*) y cuáles son los de prueba (*testSize*), y se especifica si se trata de etiquetar datos que no poseen etiquetas (*virgin* = *FALSE*). Se entrena el modelo especificando: el tipo de clasificador, en este caso árboles de decisión (*TREE*). Por último se aplica el modelo sobre los datos de prueba y se calcula la exactitud. Estos resultados se muestran en la Ilustración 27.

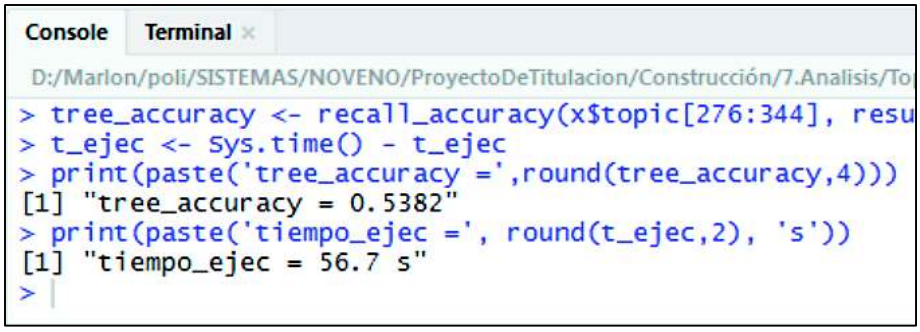


Ilustración 27. Exactitud y tiempo de ejecución del clasificador basado en árboles de decisión

Este resultado quiere decir que, del total de datos de prueba, aproximadamente el 53.82% se logró predecir de forma correcta. La ejecución tardó aproximadamente 56.7 segundos.

2.6.1.3 Máquinas de vectores de soporte

Las máquinas de vectores de soporte, en inglés *support vector machines* (SVM), es un método de clasificación basado en un discriminante, que separa el conjunto de datos basándose en la similitud de las instancias. Para encontrar el discriminante, SVM examina los datos en búsqueda de los límites entre clase y clase. Como se puede observar en la Ilustración 28.

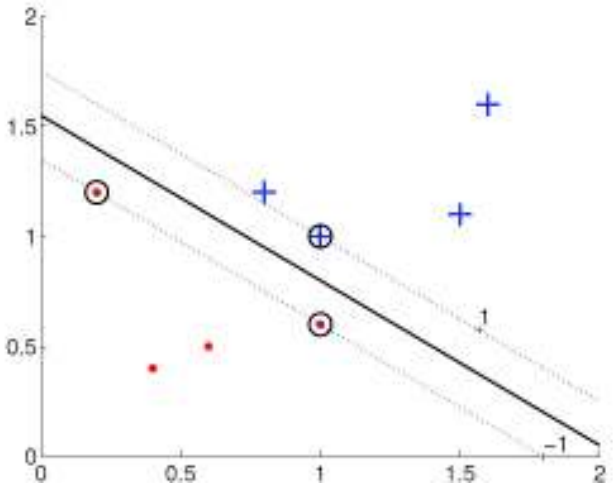


Ilustración 28. Máquinas de vectores de soporte aplicado a un problema de dos categorías (Alpaydin, 2014)

El ejemplo muestra un problema donde las instancias pertenecen a dos clases, una clase representada con cruces y la otra con puntos. La línea gruesa continua representa el límite entre clases, y las líneas punteadas a cada lado son los márgenes. El límite corresponde a un hiperplano de separación, que equidista de las instancias más cercanas entre cada clase. Para determinar los márgenes y el límite, SVM toma en cuenta solamente los datos que pertenecen a las fronteras de las clases, esto hace de SVM un modelo altamente eficiente y potente (Alpaydin, 2014).

La aplicación de máquinas de vectores de soporte en el presente proyecto se realizó mediante un kernel [7]de tipo lineal. Otros kernel como los polinomiales o los radiales, fueron probados; sin embargo, el kernel lineal produjo los mejores resultados. La creación del modelo y cálculo de su exactitud, se muestran en el Código 13.

```
# Archivo: topic_SVM.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Inicio ejecución
t_ejec <- Sys.time()
```

[7] El kernel, en SVM, es la función utilizada para dividir el conjunto de datos en clases. Las funciones de kernel pueden ser lineales, polinomiales, radiales o sigmoidales (Gromski, Xu, Turner, & Ellis, 2015). En la Ilustración 28 se muestra un kernel lineal.

```
# Crear contenedor del conjunto de datos de prueba y entrenamiento
container <- create_container(mat, x$topic,
   trainSize=1:275,testSize=276:344, virgin=FALSE)

# Crear clasificador definiendo el tipo y el kernel
model <- train_model(container, 'SVM', kernel='linear')

# Aplicar el modelo
results <- classify_model(container, model)

# Calcular exactitud
svm_accuracy <- recall_accuracy(x$topic[276:344],
   results[,"SVM_LABEL"])

# Fin ejecución
t_ejec <- Sys.time() - t_ejec

# Mostrar exactitud y tiempo de ejecución
print(paste('svm_accuracy =',round(svm_accuracy,4)))
print(paste('tiempo_ejec =', round(t_ejec,2), 's'))
```

Código 13. Cálculo de la exactitud y tiempo de ejecución de un clasificador basado en máquinas de vectores de soporte

El caso de un clasificador basado en máquinas de vectores de soporte, es similar en la implementación que uno basado en árboles de decisión. Solamente se debe definir el tipo de clasificador como "*SVM*" en la creación del modelo. Y en el cálculo de la exactitud, se debe especificar que se tomen las etiquetas estimadas por máquinas de vectores de soporte *"SVM_LABEL"*. La exactitud obtenida se muestra en la Ilustración 29.

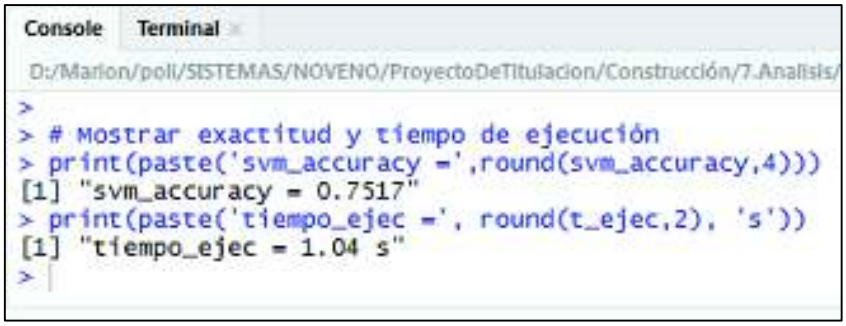


Ilustración 29. Exactitud y tiempo de ejecución del clasificador basado en máquinas de vectores de soporte

El modelo de clasificación de máquinas de vectores de soporte, logró obtener aproximadamente un 75.17% de etiquetas correctamente estimadas. Además, la ejecución tardó alrededor de 1.04 segundos.

2.6.1.4 Selección del clasificador

En base a las pruebas de los tres tipos de clasificadores, se determinó que Naïve Bayes es el clasificador de menor exactitud para nuestro caso. El clasificador basado en árboles de decisión tiene la segunda mejor exactitud, pero tiene el mayor tiempo de ejecución. Mientras que el clasificador de máquinas de vectores de soporte, logró la mejor exactitud y con el mejor tiempo. Estos resultados se pueden evidenciar en el gráfico comparativo de la Ilustración 30.

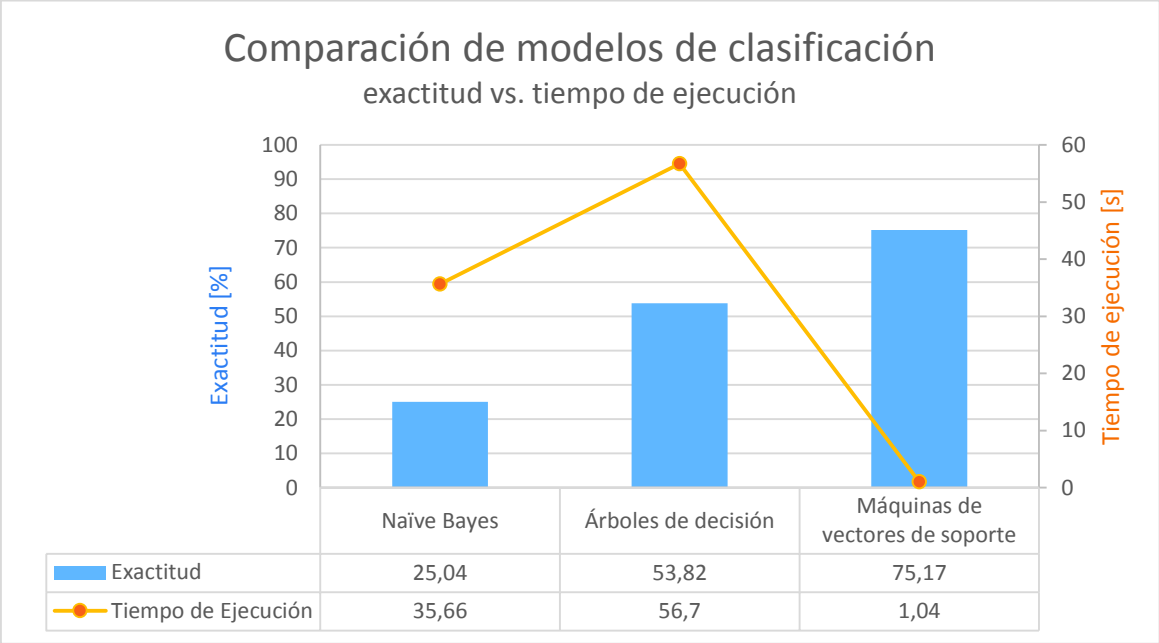


Ilustración 30. Gráfico comparativo de los modelos de clasificación, exactitud vs. tiempo de ejecución

Debido a esto, se seleccionó el clasificador basado en máquinas de vectores de soporte. La implementación del clasificador se llevó a cabo con todos los datos etiquetados manualmente como conjunto de entrenamiento; es decir, el 20% del total de datos. Las etiquetas del restante 80%, serán estimadas con el modelo de clasificación generado. El código de creación del clasificador y su aplicación sobre datos no etiquetados se muestra en el Código 14.

```
# Archivo: topic_SVM.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Crear contenedor del conjunto de datos de prueba y entrenamiento
container <- create_container(mat, x$topic,
   trainSize=1:344,testSize=345:nrow(x), virgin=TRUE)
```

```
# Crear clasificador definiendo el tipo y el kernel
model <- train_model(container, 'SVM',kernel='linear')

# Aplicar el modelo
results <- classify_model(container, model)

# Construir dataframe con etiquetas nuevas
svm_results_df <- data.frame(
  'date'=x$date,
  'source'=x$source,
  'topic'=c(as.character(x$topic[1:344]),
   as.character(results[,"SVM_LABEL"])),
  'text'=as.character(x$text))
```

Código 14. Etiquetado de datos usando un clasificador basado en máquinas de vectores de soporte

El Código 14 y el Código 13 se diferencian, básicamente, en dos sentencias. La primera es el cambio en la configuración al crear el contenedor; se debe especificar que el modelo va a trabajar con datos no etiquetados, por lo que el atributo *virgin*, es pasado como verdadero (*TRUE*). Y en la sentencia final, donde se construye un *dataframe* con los mismos campos del *dataframe* original, pero con etiquetas manuales asignadas, como se puede ver en la Tabla 8, pero añadiendo las nuevas etiquetas estimadas por nuestro modelo. El resultado es el conjunto completo de datos (1721 documentos), todos con etiquetas de la temática (*topic*) a la que pertenecen; un extracto de esto se muestra en la Tabla 9.

	date	source	topic	text
1712	2016-01-18	fb	Organización	Buen dia vecin s y buena semana Les esperamos en el en...
1713	2017-10-02	fb	Organización	Y llegaron los muebles Gracias a Buenavida Taller de Ide...
1714	2017-07-11	fb	Otro	Caminar
1715	2014-08-23	tw	Organización	La Lola Dan SXH Elysalgadoc me perdi Donde sabes si es...
1716	2017-05-13	fb	Organización	Maniana todos a la mingaaa EL COMITE PRO MEJORAS ...
1717	2016-09-08	fb	Arte	Llenaste el pasaporte De La Floresta durante tu visita a l...
1718	2013-11-25	fb	Organización	Estimados amigos y amigas Pedimos su ayuda para enco...
1719	2012-05-03	fb	Arte	TODOS INVITADOS Feria de Hortalizas Organicas en el M...
1720	2014-08-26	tw	Organización	CarlaCevallosR La iglesia de La Floresta en peligro Nadie...
1721	2017-08-22	fb	Arte	Ya se viene la Feria de Verano Agenda este Sabado y Do...

Tabla 9. Extracto del conjunto de datos etiquetado automáticamente

2.6.2 Descripción de datos

2.6.2.1 Tablas de frecuencias

Una tabla de frecuencias, es una representación de un conjunto de valores agrupados en clases o subconjuntos, junto a su correspondiente frecuencia o número de ocurrencias. Los datos pueden ser categóricos o cuantitativos. En el primer caso, las clases corresponden a un solo atributo o un conjunto de atributos. En el segundo, las clases son valores puntuales o rangos de valores. Para el caso de clases que corresponden a rangos de valores, el punto medio", puede usarse como representante de la clase. En una tabla de frecuencias, todos los valores del conjunto deben pertenecer a una y solamente una de las clases (Brase & Brase, 2013).

Las tablas de frecuencias, en este proyecto, se utilizaron para contabilizar principalmente la cantidad de documentos según sus fuentes y según las temáticas a las que pertenecen. Esto se puede realizar, una vez que se hayan etiquetado todos los documentos del conjunto de datos con sus respectivas temáticas.

La librería base de R, que es la que viene instalada por defecto, provee dos funciones con las que se construyeron las tablas de frecuencias en el proyecto. Estas funciones son *table* y *prop.table*. La primera recibe como parámetro la columna de una tabla o *dataframe* y agrupa sus elementos entre iguales, para realizar un conteo de elementos por cada clase o lo que se conoce como frecuencias absolutas. La segunda función, *prop.table*, recibe como parámetro una tabla de frecuencias absolutas y a partir de ella construye una de frecuencias relativas, o lo que es lo mismo, una tabla que lleva el porcentaje de elementos agrupados por clase.

La función *table*, se utilizó para calcular el número de posts según su fuente. El ejemplo en código y el resultado se observan en la Ilustración 31.

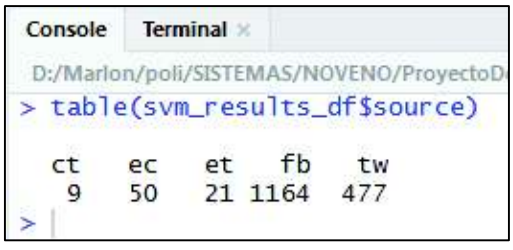


Ilustración 31. Tabla de frecuencias absolutas de las publicaciones según sus fuentes

Como se observa, se ha contabilizado el número de publicaciones alojadas en el *dataframe* de la Tabla 9, agrupados según sus fuentes (source). En la Ilustración 31, al igual que en las tablas del conjunto de datos, los códigos *ct*, *ec*, *et*, *fb* y *tw* corresponden a las fuentes The Culture Trip, El Comercio, El Telégrafo, Facebook y Twitter respectivamente. En la Ilustración 32 se puede visualizar los datos de la tabla 8 en forma de histograma.

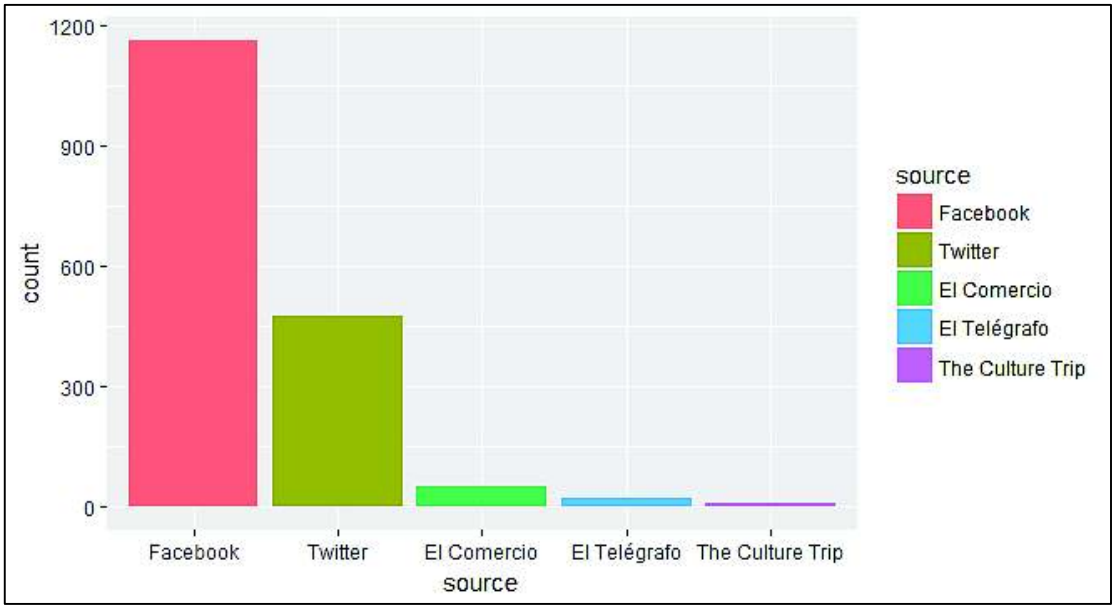


Ilustración 32. Histograma de frecuencias absolutas: cantidad de documentos según la fuente.

La Ilustración 32 (donde *source* es la fuente y *count* el número de posts) muestra que Facebook es la mayor fuente de datos en base a la cantidad de documentos, y la fuente que menos ha aportado con datos es The Culture Trip.

A su vez, la función *prop.table* se utilizó en la aplicación para determinar los porcentajes de posts según sus temáticas. El ejemplo de su aplicación se muestra en la Ilustración 33.

```
Console  Terminal
D:/Marlon/poli/SISTEMAS/NOVENO/ProyectoDeTitulacion/Construcción/8.Visualizacion/ShinyDashboard/
> abs_freq <- table(svm_results_df$topic)
> rel_freq <- prop.table(abs_freq)
> round(rel_freq * 100, 1)

    Ambiente         Arte  Gastronomia  Inseguridad    Movilidad Organización         Otro
         3.7         29.4          9.4          1.7          5.5         47.5          2.7
>
```

Ilustración 33. Tabla de frecuencias relativas de las publicaciones según sus temáticas

En este caso, como se muestra en la Ilustración 33, primero se calcula la tabla de frecuencias absolutas (*abs_freq*) de las publicaciones según sus tópicos, para a partir de ésta, generar la tabla de frecuencias relativas (*rel_freq*). El último paso es mostrar las frecuencias en forma de porcentajes y con un solo dígito decimal, a través de la función round. Esta información, representada mediante un histograma, se muestra en la Ilustración 34.

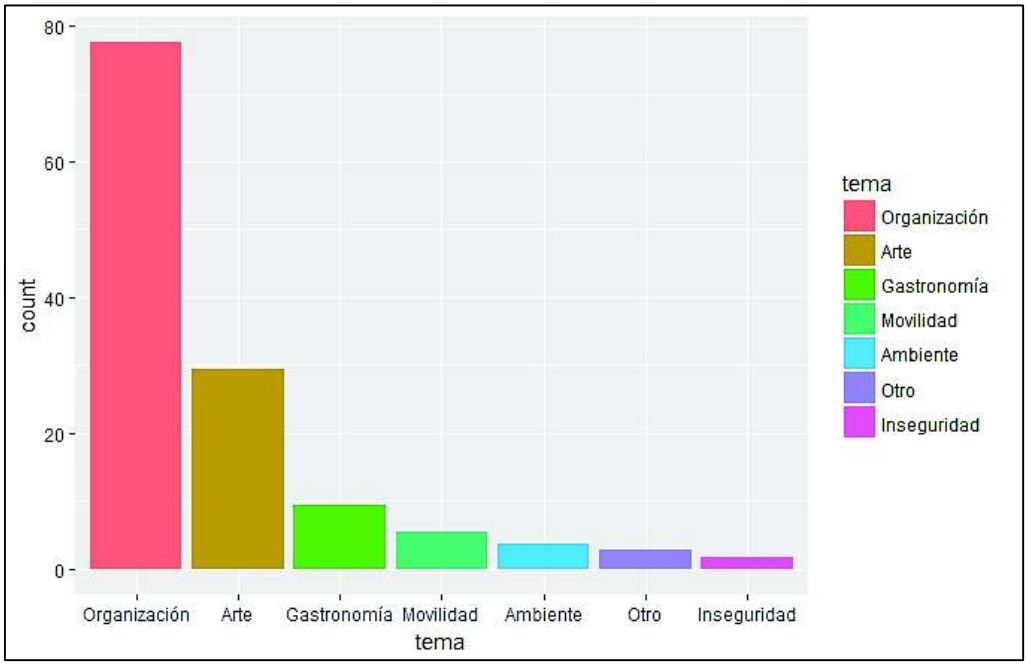


Ilustración 34. Histograma de frecuencias relativas: cantidad de documentos por temática.

De gráficos, como el de la Ilustración 34, se pueden sacar conclusiones como la identificación de las clases con mayor número de documentos, en este caso la temática Organización, seguida del Arte y de la Gastronomía; y las clases con menos documentos, en este caso la Inseguridad, seguida de la categoría Otro, Ambiente y luego Movilidad.

A partir de estas tablas, se pueden generar gráficos con la capacidad de proveer de una visión general del conjunto de datos y sus características. Más técnicas de representación gráfica de datos se detallan en la fase de 2.7.Visualización de datos.

2.6.2.2 Análisis de sentimientos

Es el método de minería de datos de texto, que procesa documentos para analizar sentimientos, opiniones, apreciaciones, actitudes y emociones de la gente hacia entidades,

que pueden ser organizaciones, servicios, productos, personas, problemáticas o tópicos. La RAE (Real Academia Española) (2014), define sentimiento como un "estado afectivo del ánimo". A su vez, la emoción es un sentimiento más intenso y de corta duración, causado por un objeto en concreto, como un suceso, una persona, una cosa o un tópico. Las oraciones que expresamente muestran un sentimiento, generalmente son subjetivas, a diferencia de las oraciones que plantean hechos. Sin embargo, las oraciones objetivas pueden comunicar hechos deseables y no deseables, que implican sentimientos positivos y negativos en sus autores o en los receptores de dicho mensaje (Liu, 2015).

Una librería de R que permite, entre otras cosas, extraer sentimientos y emociones a partir de texto, es *Syuzhet*, la cual utiliza una variedad de diccionarios, entre ellos *NRC Word-Emotion Lexicon* (CRAN Project, 2017). El diccionario de NRC es una lista que actualmente contiene 14182 etiquetas asignadas entre 6468 palabras diferentes, que corresponden a la polaridad del sentimiento y la emoción que expresan. NRC ofrece versiones de su diccionario para aproximadamente cuarenta idiomas incluido el español, ayudándose de *Google Translate* para la traducción de palabras (Mohammad & Turney, 2017). Las etiquetas asignadas son puntuaciones; marcadores dicotómicos que especifican si una determinada palabra corresponde o no a cada tipo de emoción o sentimiento. Para determinar las emociones y polaridad sentimental, se toma cada palabra del texto a analizar que existe en el diccionario y se marca con 1 en cada tipo de emoción a la que corresponde, y con 0 en aquellas a las que no; de igual manera, son marcadas con 0 las palabras que no existen en el diccionario. Al final se determinan las emociones mediante tablas de frecuencias absolutas (Munezero, Montero, & Mozgovoy, 2015), similar a la Ilustración 31.

La determinación de sentimientos y emociones del conjunto de textos recogidos se puede realizar mediante la función *get_nrc_sentiment* de la librería *Syuzhet*, como se muestra en Código 15.

```
# Archivo: sent_analysis.R
# Autor: Marlon Vargas
# Fecha: noviembre 2017

# Crear de matriz de emociones y sentimientos
emotion_mat <- get_nrc_sentiment(svm_results_df$text, language =
   "spanish")

# Crear tabla de frecuencias absolutas
transp <- data.frame(t(emotion_mat))
abs_freq <- data.frame("count" = rowSums(transp[]))
```

```
# Graficar emociones
qplot(sentiment, data=abs_freq[1:8,], weight=count,
   geom="bar",fill=sentiment)

# Graficar sentimientos
qplot(sentiment, data=abs_freq[9:10,], weight=count,
   geom="bar",fill=sentiment)
```

Código 15. Extracción de emociones y sentimientos de textos

En primer lugar, *get_nrc_sentiment* construye una matriz de emociones y sentimientos, como se puede observar en la Tabla 10, la cual se forma a partir del conteo de palabras agrupadas según la emoción y el sentimiento que expresan.

	Ira	Anticipación	Disgusto	Temor	Alegría	Tristeza	Sorpresa	Confianza	Negativo	Positivo
1	0	5	0	0	5	0	2	4	0	10
2	0	3	0	2	0	0	0	0	2	0
3	0	1	0	0	1	0	1	0	0	1
4	0	1	0	0	1	0	1	0	0	1
5	0	1	0	0	1	0	1	0	0	2
6	0	0	0	0	1	0	0	0	0	2
7	0	0	0	0	0	0	0	0	1	1
8	2	4	1	4	4	0	0	5	7	7
9	0	1	0	0	1	0	1	0	0	2
10	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	2
12	0	1	0	0	0	0	0	0	0	2

Tabla 10. Extracto de matriz de emociones y sentimientos

El análisis basado en diccionarios de sentimientos considera cada documento como una bolsa de palabras, en donde a cada una se le dio una valoración según la emoción y sentimiento que expresa, y según esto agruparlas y contabilizarlas. Como se observa en la Tabla 10. Los sentimientos, en el diccionario NRC, se clasifican en positivos y negativos. Y las emociones se clasifican en ocho, que corresponden a las emociones básicas y necesarias para la supervivencia, según Plutchik (2003), estas son: ira, anticipación, desagrado, miedo, alegría, tristeza, sorpresa y confianza.

A partir de la Tabla 10, se puede generar una tabla de frecuencias absolutas. Ésta nos permite saber el número de palabras totales del corpus (*count*), agrupadas por emociones (*emotions*) o sentimientos (*sentiment*), como se observa en la Tabla 11.

emotion	count
Ira	683
Anticipación	1715
Disgusto	594
Temor	1006
Alegría	1552
Tristeza	1046
Sorpresa	741
Confianza	2540

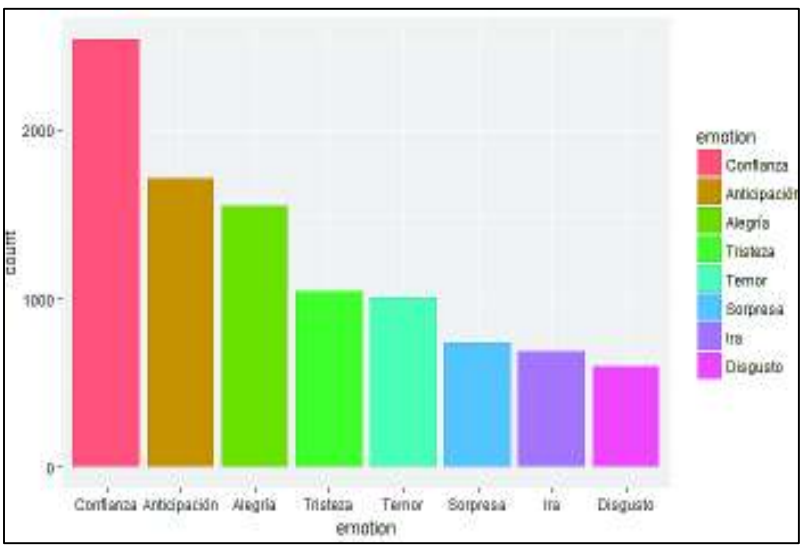
a)

sentiment	count
Negativo	2075
Positivo	5161

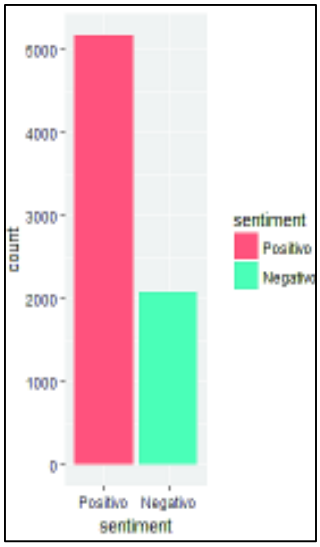
b)

Tabla 11. Tabla de frecuencias absolutas de palabras según a) emociones y b) sentimientos

De la Tabla 11, se pueden graficar los resultados mediante histogramas, como los que se muestran en Ilustración 35.



a)



b)

Ilustración 35. a) Histograma de emociones. b) Histograma de sentimientos.

Se identificaron las emociones y sentimientos generales expresados en los textos acerca de La Floresta. De las emociones, destaca la alegría y la anticipación; las emociones menos presentes son la ira y el disgusto. De los sentimientos, prevalecen los positivos sobre los negativos.

2.6.2.3 Modelado de Tópicos

Uno de los métodos más utilizados de *clustering*[8] para documentos de texto, es *topic modeling* o modelado de tópicos. Éste se basa en un modelo probabilístico generativo para los documentos del corpus, es decir, un modelo que define un proceso simple de probabilidad por el cual las palabras de un documento están siendo generadas en base a un pequeño número de variables latentes o tópicos. Mediante técnicas estadísticas, topic modeling invierte este proceso para lograr inferir las variables latentes responsables de generar tales documentos (Berry & Kogan, 2010).

Latent Dirichlet Allocation (LDA), es un modelo generativo ampliamente usado para estimar tópicos generados a partir de documentos. LDA recibe como entrada un corpus o conjunto de documentos, representados como bags of words, y un número de tópicos, de los cuales se asume que se encuentran distribuidos entre los documentos. LDA toma a los documentos como una mezcla de varios tópicos y a cada tópico como una variedad de términos, donde cada término contribuye al tópico del documento. Al considerar cada documento como una combinación de varios tópicos, LDA busca determinar cuáles son y en qué proporciones se encuentran dentro de cada documento (Ashish & Avinash, 2016).

En R, la librería *topicmodels* trae consigo la función *LDA*. El procedimiento para usarla empieza por construir el corpus y realizar la limpieza de texto con *tm* como se muestra en el Código 9. Se construye la matriz de frecuencias documento-término como en Código 10 y se define el número de tópicos que se desea estimar. Sea *k*, el número de tópicos a estimar y *mat*, la matriz de frecuencias de palabras. La forma de determinar los tópicos se realiza mediante el programa del Código 16.

```
# Archivo: topic_LDA.R
# Autor: Marlon Vargas
# Fecha: diciembre 2017

# Definir el número de tópicos
```

[8] *Clustering* es un método de aprendizaje no supervisado, que busca agrupar objetos basándose en regularidades o patrones en los datos de entrada. En minería de texto, el objetivo es agrupar documentos de características afines. Generalmente, la agrupación se realiza en base al número de palabras que comparten entre documentos (Alpaydin, 2014).

```
k <- num_topics

# Aplicación de LDA() sobre la matriz de frecuencias mat
lda <- LDA(mat, k)

# Extraer los términos estimados
topics <- terms(lda)
```

Código 16. Uso de LDA para estimar tópicos

El modelado de tópicos se utilizó en el documento, para extraer las palabras de mayor importancia. A pesar de que esto no fue suficiente para determinar las temáticas exactas de cada documento, sirve para dar una buena idea generalizada de lo que se está hablando en el texto. A continuación, en la Tabla 12, se muestran los resultados de aplicar LDA sobre textos correspondientes a la misma temática.

Tema	**Tópicos**				
Arte	cultural	talleres	junio	casa	feria
Otro	hoy	parque	another	plaza	juntos
Organización	parque	ciudad	iglesia	vecinos	espacio
Movilidad	silencio	velocidad	lugar	vizcaya	ciudad
Gastronomia	romolo	tacos	comida	tripas	parque
Ambiente	ecuador	arbolado	huerta	pichincha	llamas
Inseguridad	vecinos	robos	comedy	zona	cuidado

Tabla 12. Resultados de LDA sobre datos de mismas temáticas

El código responsable de generar los tópicos de la Tabla 12, se muestra en el Código 17.

```
# Archivo: topic_LDA.R
# Autor: Marlon Vargas
# Fecha: diciembre 2017

# Aplicar función a cada tema de las etiquetas
for (tema in unique(svm_results_df$topic)) {

  # Extraer datos respecto a un determinado tema
  tema_posts <- svm_results_df %>% filter(topic == tema)

  # Aplicar función creada (compuesta de tm y LDA)
  tmp_topics <- getTopics(tema_posts$text, 5)

  # Añadir tópicos encontrados en un dataframe,
  # agrupados por tema
  topic <- data.frame(
```

```
    "tema" = tema,
    "topics" = paste0(tmp_topics, collapse = ", "))
  topics <- rbind(topics, topic)
}
# Imprimir dataframe de resultados
topics
```

Código 17. LDA iterativo para obtener tópicos de varios documentos según sus temáticas

2.7 Visualización de datos

Existe una amplia variedad de gráficos, unos más sofisticados que otros, y nuevos tipos son creados todo el tiempo. Sin embargo, los gráficos más utilizados son aquellos que por su simplicidad, facilitan al usuario su comprensión. Los tipos de gráficos, poseen distintos enfoques según los tipos de preguntas que se deseen resolver mediante su interpretación (Meyer & Fisher, 2018).

Como etapa de visualización, se desarrolló un tablero de mando web. La construcción se realizó utilizando una librería de R llamada *Shiny*. Esta librería permite la creación de aplicaciones web interactivas, usando R, HTML, CSS y Javascript. Shiny combina el poder computacional de R con la interactividad de las páginas web modernas, lo que la hace una potente herramienta de visualización de datos (Shiny, 2013).

Shiny corre en el mismo entorno de desarrollo de R, R Studio. Sin embargo, para su publicación, existen dos opciones. La primera es montar un servidor Linux y configurar *Shiny Server,* que es el que maneja el servidor web donde se va a alojar nuestra aplicación. La otra opción es alojar nuestra aplicación Shiny en la nube.

Shinyapps.io es el servicio de Shiny que ofrece el hosting exclusivamente de aplicaciones desarrolladas con su librería. Como parte del objetivo del proyecto se pretendía poner el producto desarrollado al alcance del público. Es por esto que el acceso a la aplicación de minería de texto desarrollada, está disponible en la siguiente dirección URL:

https://marlon.shinyapps.io/WebTextMiningLaFloresta/

A continuación se detallan los componentes gráficos utilizados en Shiny para visualizar los resultados de la minería de texto.

2.7.1 Representación de frecuencias absolutas

Las frecuencias absolutas representan el número de apariciones de un valor en el conjunto de datos. La sumatoria de las frecuencias absolutas es igual al número total de valores del conjunto de datos (Sangaku Maths, 2018). Las frecuencias absolutas se representan generalmente mediante histogramas como el de la Ilustración 32. El principal objetivo es contabilizar objetos agrupados en clases (Meyer & Fisher, 2018).

Para brindar una visión general del número de documentos analizados, en lugar de optar por histogramas, se seleccionaron cajas de texto propias de Shiny, mostradas en la Ilustración 36, que publican los valores de las frecuencias absolutas o cantidad de documentos del conjunto de datos.

Ilustración 36. Visualización del número de documentos analizados

Esta técnica funciona ya que, en este caso, no se posee una gran cantidad de clases. Se construyeron cinco cajas, de las cuales a Facebook y a Twitter se le asignaron una a cada una debido a la cantidad de documentos que aportaron al análisis. Y se agruparon las noticias en un conjunto y los blogs en otro. Se creó una caja principal para mostrar la cantidad total de documentos procesados.

2.7.2 Representación de frecuencias relativas

Las frecuencias relativas son otra forma de representar la contabilización de elementos de un conjunto de datos. La frecuencia relativa se define como el cociente de la frecuencia absoluta entre el número de valores del conjunto de datos. La sumatoria de las frecuencias relativas es igual a 1. Éstas pueden representarse también como porcentajes (Sangaku Maths, 2018). El gráfico típico para mostrar frecuencias relativas, es el gráfico circular o de pastel. Este gráfico es una modificación del gráfico de barras, en donde el valor del atributo es representado por el ángulo de la porción en lugar de la altura de una barra. Este gráfico simboliza “las partes de un todo” (Meyer & Fisher, 2018).

El gráfico circular, en la aplicación, se utilizó para representar las frecuencias de los documentos clasificados según las temáticas identificadas, como se muestra en la Ilustración 37. El origen de los datos es el etiquetado automático de los documentos y su posterior cálculo de frecuencias relativas, como se muestra en la Ilustración 33 en la sección de 2.6.2.1.Tablas de frecuencias.

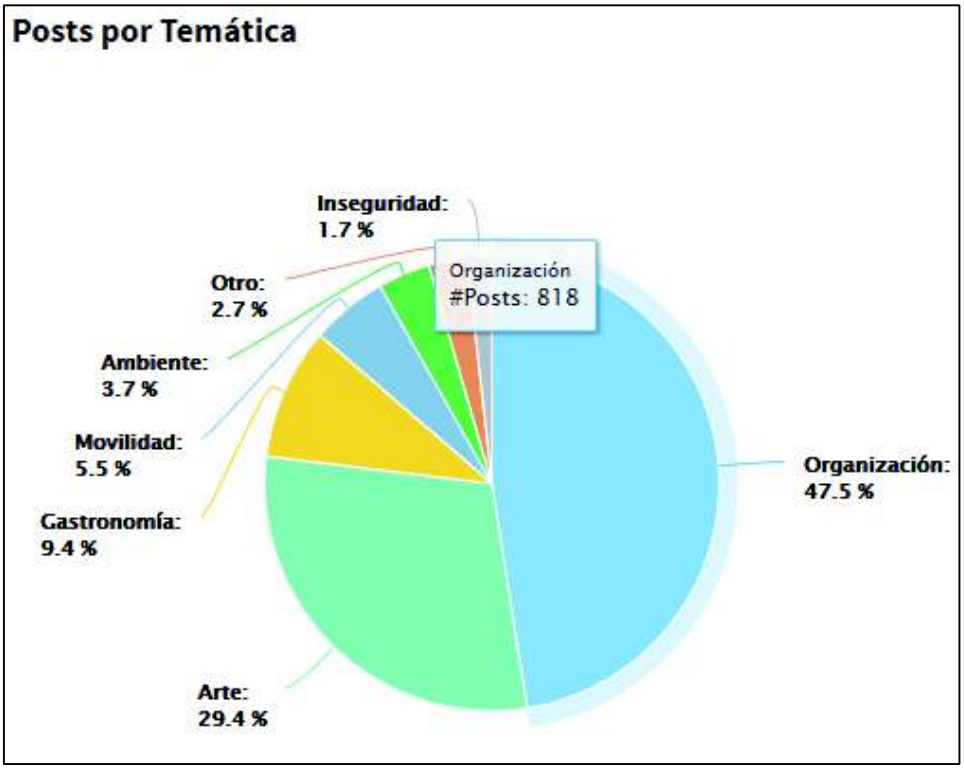


Ilustración 37. Gráfico circular de frecuencias de documentos según su temática

De este tipo de gráfico se pueden sacar conclusiones como, por ejemplo, que el tema de la Organización barrial es el tema del que más se publica en las fuentes de datos, seguido del Arte, Gastronomía, Movilidad y Ambiente; y que el tema del que menos se publica es la Inseguridad.

Para la creación del gráfico de la Ilustración 37, se utilizó *highcharter*, que es una librería de Javascript adaptada para utilizarse con R y Shiny, para generar gráficos web interactivos.

2.7.3 Nube de palabras

En minería de texto, visualizar documentos es una tarea necesaria. Al no existir una técnica definitiva de visualización de texto, la mayoría de métodos caen dentro de un enfoque similar. Uno de los enfoques más utilizados se basa en el conteo de las frecuencias de palabras o frases, y representar dichos datos mediante gráficos de barras, líneas o nubes de palabras. Las nubes de palabras o *word clouds*, son otro tipo de gráficos que actualmente se utilizan en gran cantidad dentro de analítica de texto. Al igual que un gráfico de barras o un histograma, el conteo de ítems es representado mediante la variación del tamaño de los componentes del gráfico; en el caso de las nubes de palabras, los tamaños de fuente son proporcionales a las frecuencias de las palabras que contiene un texto (Meyer & Fisher, 2018).

Ilustración 38. Nube de palabras de publicaciones sobre La Floresta

La nube de palabras se crea a partir de la matriz de frecuencias de palabras, que se construye a mediante el Código 10. Los datos utilizados para generar la nube de palabras de la Ilustración 38, corresponden a todos los textos de las publicaciones históricas recogidas. De este gráfico se puede concluir, por ejemplo, que vecinos es la palabra más común entre las publicaciones, seguida de sábado, espacio, cultural, feria, parque y municipio, entre otras palabras usadas con frecuencia.

La aplicación tiene la capacidad de generar nubes de palabras según la temática. Esto se muestra más adelante en la presentación de la aplicación, en el capítulo 3.RESULTADOS Y DISCUSIÓN.

2.7.4 Mapa geográfico

Los mapas geográficos utilizan variedades de colores y símbolos para representar datos referentes a lugares. Este tipo de gráficos se utilizan para simbolizar similitudes y variaciones entre regiones. Los mapas tienen la ventaja de ser muy familiares para las personas, muchas veces no se requiere el etiquetado para que un usuario sepa cuál es el lugar en cuestión (Meyer & Fisher, 2018).

En el proyecto, se utilizaron mapas para ubicar los sitios turísticos de Quito y con esto lograr comparaciones, como se muestra en la Ilustración 39. Se pretende determinar la concentración de lugares de interés para tener una idea de la aceptación comercial que puede tener La Floresta en relación con otros sectores de la ciudad.

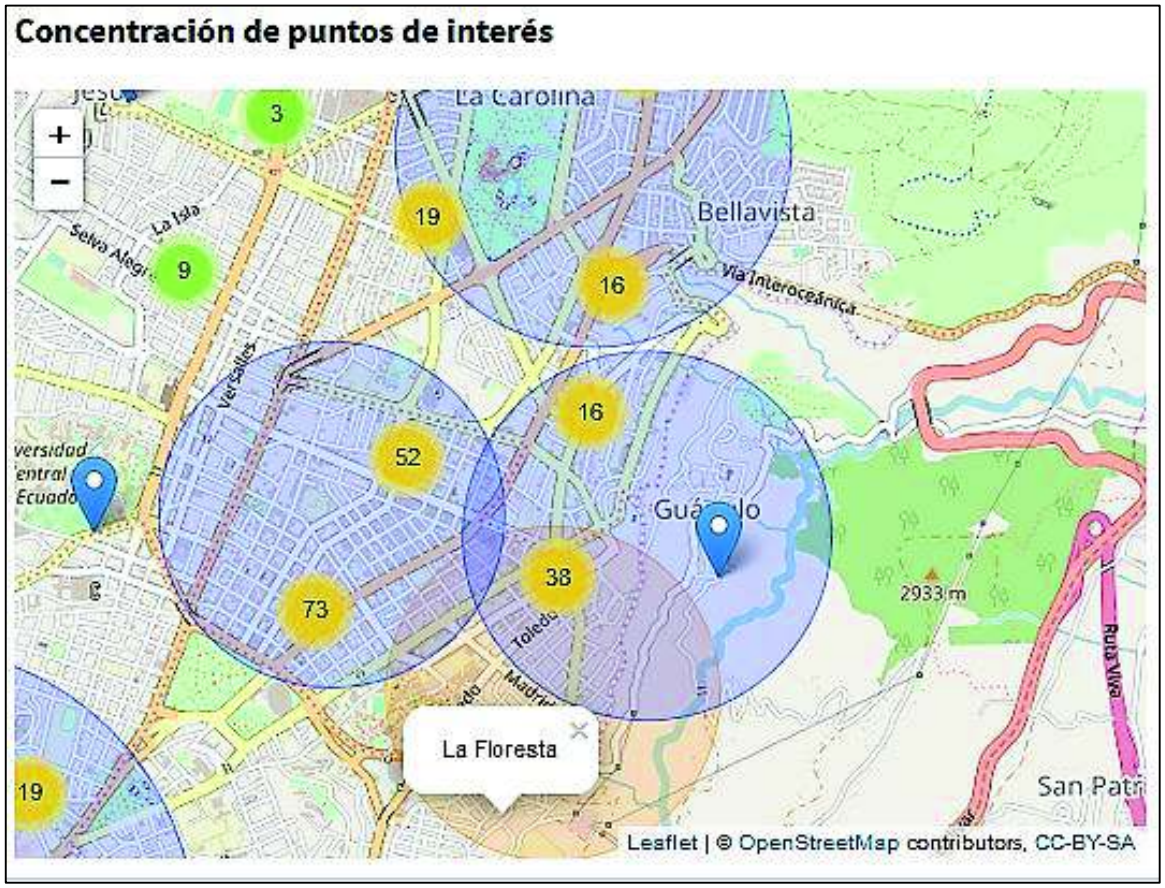


Ilustración 39. Mapa de concentración de sitios turísticos de interés, sector La Mariscal

La Ilustración 39 muestra mediante burbujas las agrupaciones según el número de sitios de interés, las de color amarillo corresponden a las de mayor número y las verdes a las de menor número. En el gráfico se puede observar que el mayor número de sitios de interés se encuentran cerca al centro de La Mariscal, seguido de La Floresta junto a Guápulo y finalmente el sector de La Carolina.

Se ubicaron también los sitios turísticos según su popularidad y junto a sus calificaciones extraídas de Trip Advisor. Este y otros resultados se ven con mayor detalle en la siguiente sección de RESULTADOS Y DISCUSIÓN.

3. RESULTADOS Y DISCUSIÓN

3.1 Resultados

El producto final del desarrollo del proyecto es un sistema de análisis de datos textuales de las fuentes web: El comercio, El Telégrafo, Facebook, Twitter, Trip Advisor y The Culture Trip. El sistema realiza un tratamiento de los datos, de tal forma que asegura su validez; estos son introducidos a varios procesos de análisis, con el fin de transformar el texto en ideas. Después del análisis, se tendrán documentos clasificados según las temáticas que tratan junto a los sentimientos y emociones que expresan. Los resultados de dichos análisis son presentados finalmente mediante un tablero de mando de acceso público.

A continuación, se presenta la aplicación, sus funcionalidades y los resultados que presenta del análisis.

3.1.1 Presentación de la aplicación

El enlace de acceso público al tablero de mando es:

https://marlon.shinyapps.io/WebTextMiningLaFloresta/

La interfaz de usuario de la aplicación hace uso de la estructura básica de un tablero de mando, como se indica en la Ilustración 40, en la que se tiene: una barra superior de título; una barra lateral izquierda, donde se tienen los controles de usuario de la aplicación o entradas; y una sección principal donde se muestran los resultados del análisis o salidas, dispuestos éstos a actualizarse en cuanto los controles de usuario los hagan cambiar.

Ilustración 40. Estructura del tablero de mando

El tablero de mando consta dos páginas principales: el reporte general y el detalle.

Reporte general

Ilustración 41. Tablero de mando: Página de reporte general

En la página de la Ilustración 41 se muestran los resultados generales del análisis: el número de posts totales y según su fuente; los porcentajes de posts según las temáticas; una nube de palabras correspondiente a todo el conjunto de textos; un gráfico de barras correspondiente al análisis de sentimientos por temática; y un mapa de Quito con la concentración de puntos de interés. A continuación se muestran los controles de usuario y los gráficos que controlan:

- Una entrada de rango de fechas que controla el intervalo de tiempo en el que se quieren observar los resultados. Este control tiene efecto sobre todos los gráficos excepto sobre el mapa, debido a que los datos de ubicación geográfica provenientes de Trip Advisor no poseen información de fechas de publicación. En la Ilustración 42 también se observa el gráfico circular, con el porcentaje de documentos según cada temática.

Ilustración 42. Tablero de mando: Entrada de rango de fechas

- Un deslizador que define el número de palabras a mostrar en la nube de palabras. Esto sirve para averiguar distintas cantidades de palabras significativas encontradas en el texto. Esto se observa en la Ilustración 43.

Ilustración 43. Tablero de mando: controlador de nube de palabras

- Un par de botones de selección que permiten escoger entre visualizar el análisis por sentimientos o por emociones. Esto se muestra en la Ilustración 44.

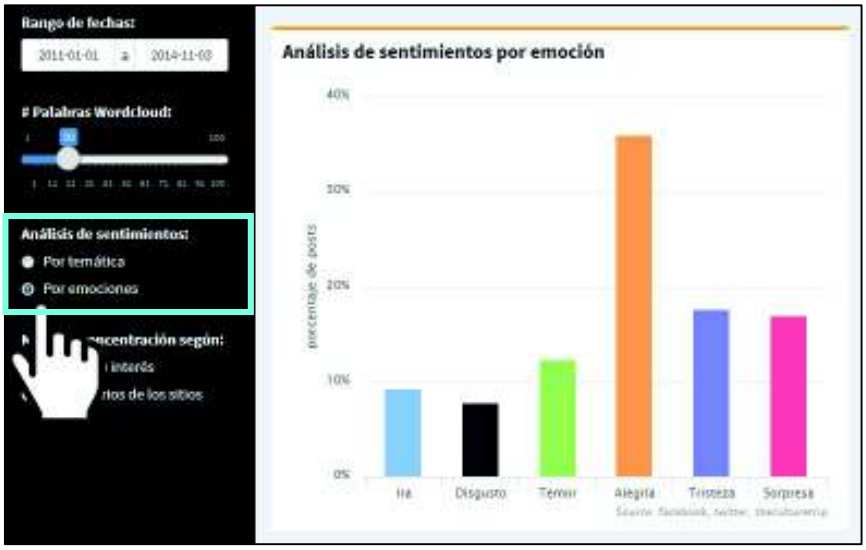


Ilustración 44. Tablero de mando: Controlador de gráfico de sentimientos y emociones

- Y otro par de botones de selección que permiten escoger el tipo de mapa a mostrar, de entre puntos de interés o según los comentarios y popularidad de los sitios turísticos. Esto se puede ver en la Ilustración 45.

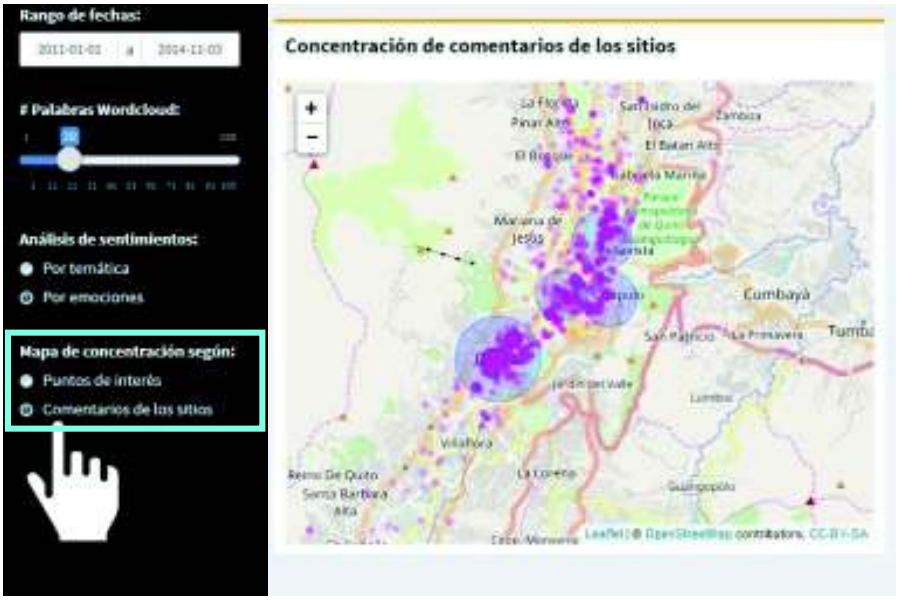


Ilustración 45. Tablero de mando: Controlador de tipo de mapa

Detalle

En esta página se muestran los resultados del análisis, organizados por temática. Al ingresar por primera vez, se desplegará el mensaje que se muestra en la Ilustración 46.

Ilustración 46. Tablero de mando: Mensaje inicial de la página de detalle

En la página de detalle se presenta un control de usuario adicional, que se indica en la Ilustración 47, éste sirve para seleccionar la temática sobre la que se quiere observar los resultados del análisis.

Ilustración 47. Tablero de mando: Selección de temática para mostrar detalle

Los controles de rango de fechas y de número de palabras de la nube tienen la misma funcionalidad que en la página del reporte general. Por otro lado, la página de detalle presenta diferencias respecto a sus gráficas. Las gráficas de la página de detalle son:

Ilustración 48. Tablero de mando: Página de detalle

1) Caja informativa de la cantidad de documentos analizados dentro del rango de tiempo especificado.
2) Gráfico de una barra con el porcentaje de documentos de sentimientos positivos, negativos y neutrales.
3) Top 5 de los tópicos más representativos respecto a la temática.
4) Nube de palabras que es controlada por el deslizador de la barra izquierda.
5) Gráfico de análisis de emociones.

3.1.2 Resultados del análisis

3.1.2.1 Resultados generales

Se determinó que las temáticas de las que más se hablan en la web respecto a La Floresta, de mayor a menor son: la organización barrial, con un 47.5%; el arte, con un 29.4%; y la gastronomía, con 9.4%. Mientras que las temáticas con menor publicaciones son: la movilidad, con 5.5%; el medio ambiente, con un 3.7%; de otros temas, un 2.7%; y de la que menos se habla es la inseguridad, con 1.7%. Esta información se muestra en el gráfico de la Ilustración 37. Gráfico circular de frecuencias de documentos según su temática.

El análisis de sentimientos reveló que el tema del medio ambiente es del que más positivamente se habla, seguido del arte, y la movilidad. El tema de la inseguridad es del que más negativamente se habla, después le sigue la movilidad y la organización. La gastronomía es el tema que menos negatividad tiene, al mismo tiempo, es del que más se habla neutralmente. Esto se puede observar en la Ilustración 49. Gráfico de análisis de sentimientos por temática.

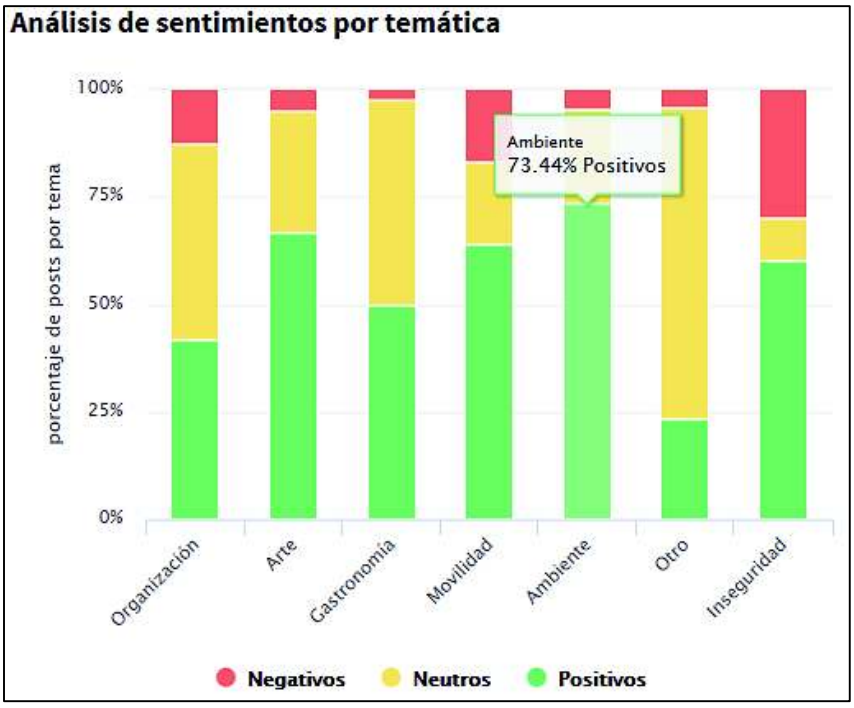


Ilustración 49. Gráfico de análisis de sentimientos por temática

Entre las palabras más mencionadas en las publicaciones se tienen: vecinos, sábado, espacio, feria, cultural, parque, público, municipio, arte, entre otras. Éstas pueden

observarse en la Ilustración 38. Nube de palabras de publicaciones sobre La Floresta. Este conjunto de palabras demuestra la unión de la gente del barrio para organizarse, así como su apoyo a las actividades artísticas.

La ubicación de los sitios turísticos, según datos de Trip Advisor (2017), en el mapa de Quito demostró que las mayores concentraciones se encuentran principalmente en tres sectores: el Centro Histórico, La Carolina y La Mariscal, parroquia que contiene al barrio La Floresta. La mayor concentración de negocios de La Floresta, registrados en Trip Advisor, se encuentra en las zonas más cercanas a Guápulo y la Av. González Suárez. Hacia el norte existen algunos puntos de interés registrados y en el sur la cantidad es mínima. Esto se puede evidenciar en la Ilustración 39. Mapa de concentración de sitios turísticos de interés, sector La Mariscal y en la Ilustración 50. Mapa de concentración de sitios turísticos según su popularidad.

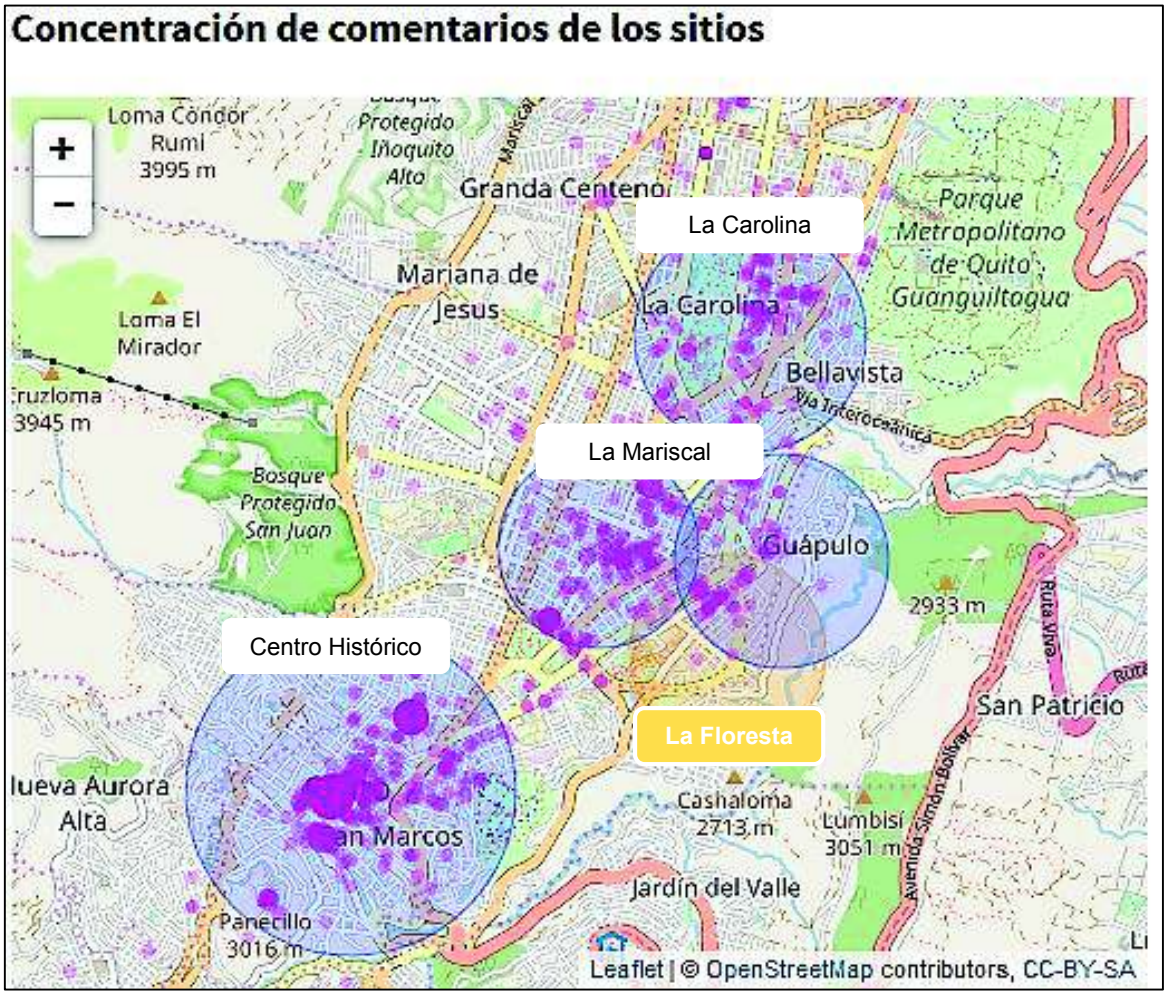
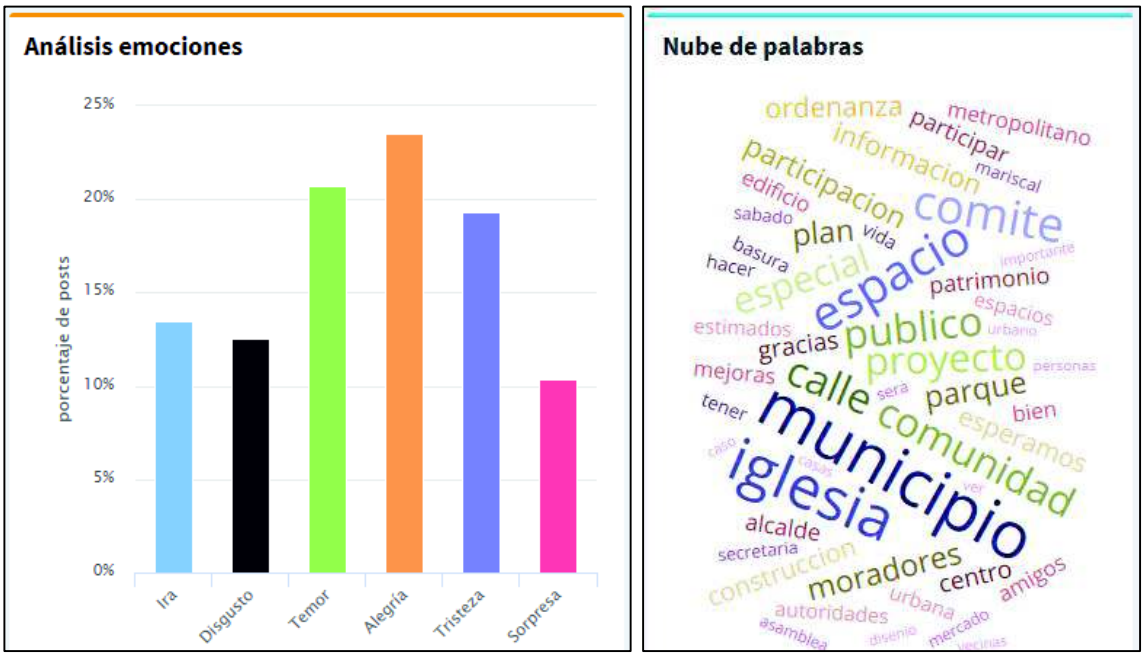


Ilustración 50. Mapa de concentración de sitios turísticos según su popularidad

3.1.2.2 Resultados por temática Organización

Respecto a la organización, que es el tema más hablado, 41.69% de las publicaciones son positivas, 12.84% son negativas y tiene un 45.48% de publicaciones que no expresan una polaridad de sentimientos, como se muestra en la Ilustración 49. Gráfico de análisis de sentimientos por temática.



a) b)

Ilustración 51. La Floresta, análisis por temática organización. a) Histograma de emociones b) Nube de palabras

Las emociones expresadas, con respecto a Organización, son diversas. Sin embargo, 23.55% corresponde a publicaciones que expresan alegría, seguida del 20.7% de temor, un 19.35% de tristeza, 13.51% de ira, 12.55% de disgusto y 10.33% de sorpresa, como se puede ver en la Ilustración 51.a.

Los tópicos o palabras más utilizadas fueron: municipio, comunidad, iglesia, vecinos y comité, como se ve en Ilustración 51.b. En estos tópicos se nota la unión de los vecinos del barrio para organizarse y hacerse escuchar ante las autoridades, en este caso el municipio de Quito.

3.1.2.3 Resultados por temática Arte

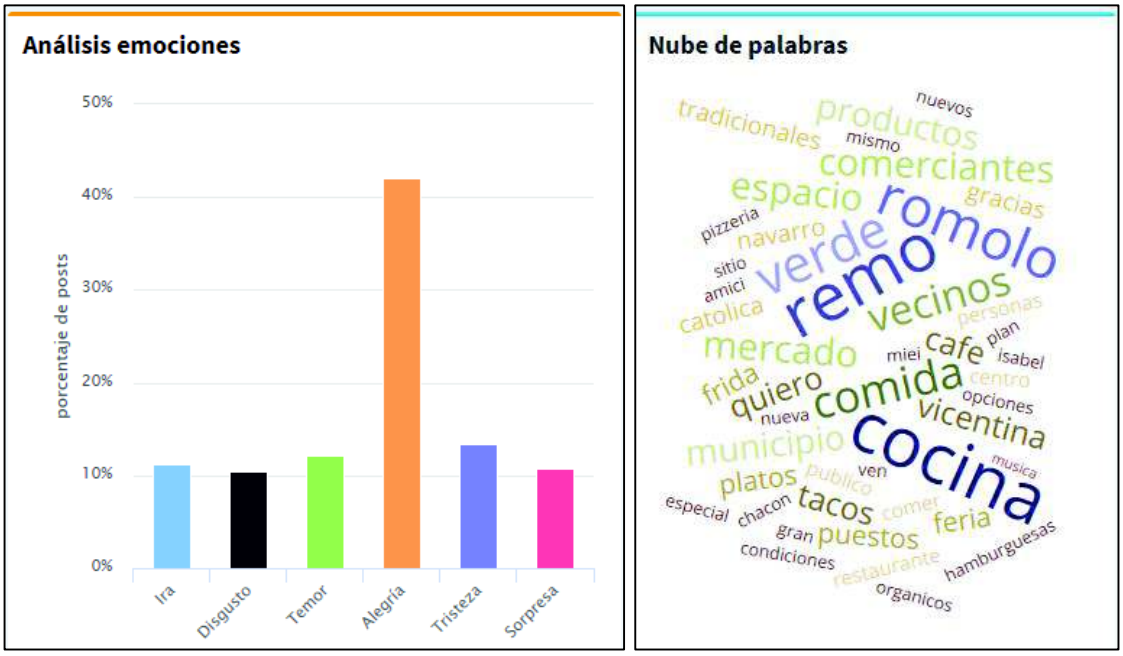
a) *b)*

Ilustración 52. La Floresta, análisis por temática arte. a) Histograma de emociones b) Nube de palabras

Del arte, 66.8% se habla positivamente, 4.94% negativamente y tiene un 28.26% de publicaciones neutrales, como se ve en la Ilustración 49. Gráfico de análisis de sentimientos por temática. De las emociones, alegría obtuvo un 35.49%, seguido de sorpresa con un 18.56%, un 17.28% de tristeza, 12.16% de temor, y alrededor de un 16% entre ira y disgusto, como se muestra en la Ilustración 52.a.

Los tópicos detectados en la Ilustración 52.b fueron: cultural, verano, ochoymedio, música y feria. Donde se mencionan palabras como “verano”, que hace referencia al espacio artístico “Feria de verano”; y “ochoymedio” hace referencia a un cine popular del barrio, llamado Ocho y Medio.

3.1.2.4 Resultados por temática Gastronomía



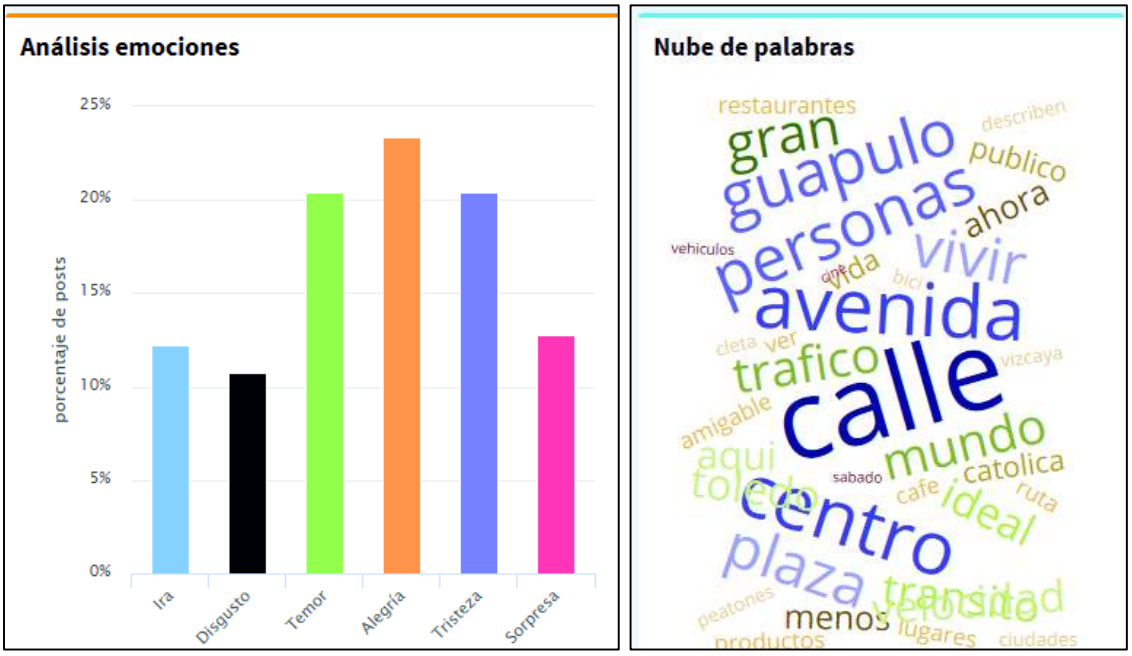
a) *b)*

Ilustración 53. La Floresta, análisis por temática gastronomía. a) Histograma de emociones b) Nube de palabras

De gastronomía se habla un 50% positivamente, un 2.47% negativamente y el resto fueron neutrales, como se ve en la Ilustración 49. Gráfico de análisis de sentimientos por temática. La alegría fue la emoción más expresada con un 41.99%, seguida de sorpresa con un 13.42%, ira un 11.26%, disgusto un 10.82%, y el resto entre temor y tristeza, como se muestra en la Ilustración 53.a.

Los tópicos encontrados fueron: Romolo, cocina, comida, parque y verde; lo que se muestra en la Ilustración 53.b. Donde se mencionan palabras como “Romolo”, que hace referencia a un restaurante de comida italiana llamado “Romolo e Remo”; “parque”, que se refiere al tradicional parque de La Floresta donde por las noches se vende comidas típicas; y “verde”, que hace referencia a “Te Quiero Verde” , una tienda de productos alimenticios artesanales.

3.1.2.5 Resultados por temática Movilidad



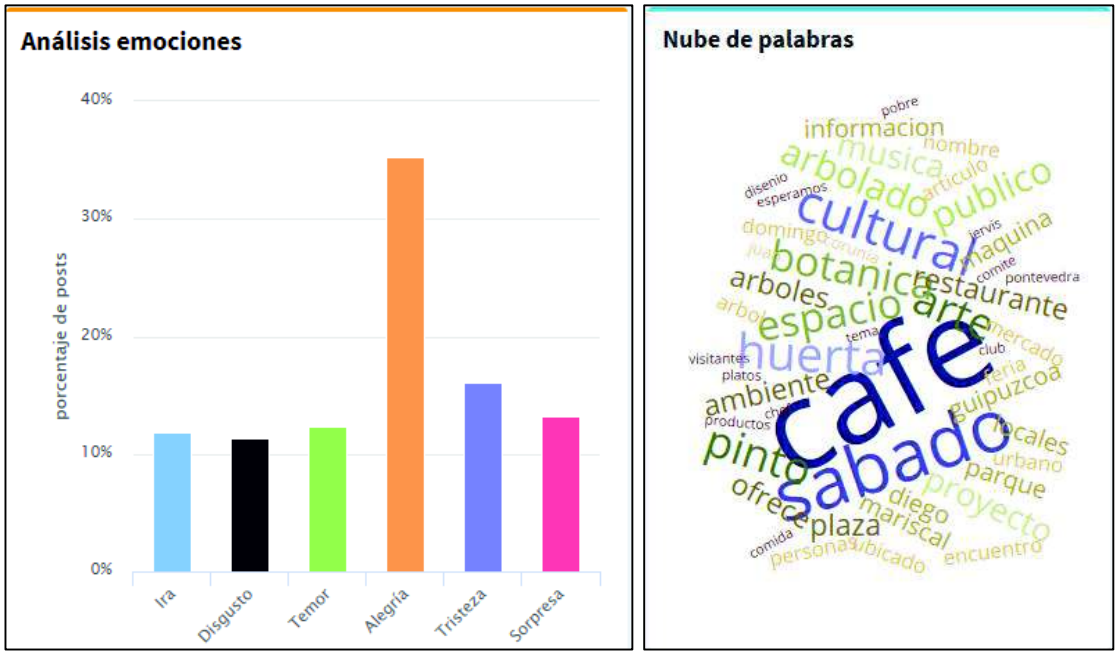
a) *b)*

Ilustración 54. La Floresta, análisis por temática movilidad. a) Histograma de emociones b) Nube de palabras

Acerca de movilidad se habló un 63.83% positivamente, un 17.02% negativamente y el resto de neutros, como indica la Ilustración 49. Gráfico de análisis de sentimientos por temática. Se muestra entre temor, tristeza y alegría un nivel similar, de alrededor de 21% de publicaciones cada uno. Seguido de un 12.80% de sorpresa, un 12.26% de ira y un 10.74% de disgusto, como se ve en la Ilustración 54.a.

Los tópicos más relevantes encontrados fueron: calle, tráfico, centro, avenida y Guápulo. Esto se ve en la Ilustración 54.b. Donde se puede notar que se menciona a Guápulo, debido al tráfico que se produce por los automóviles que pasan por allí para llegar a la Av. Simón Bolivar.

3.1.2.6 Resultados por temática Ambiente



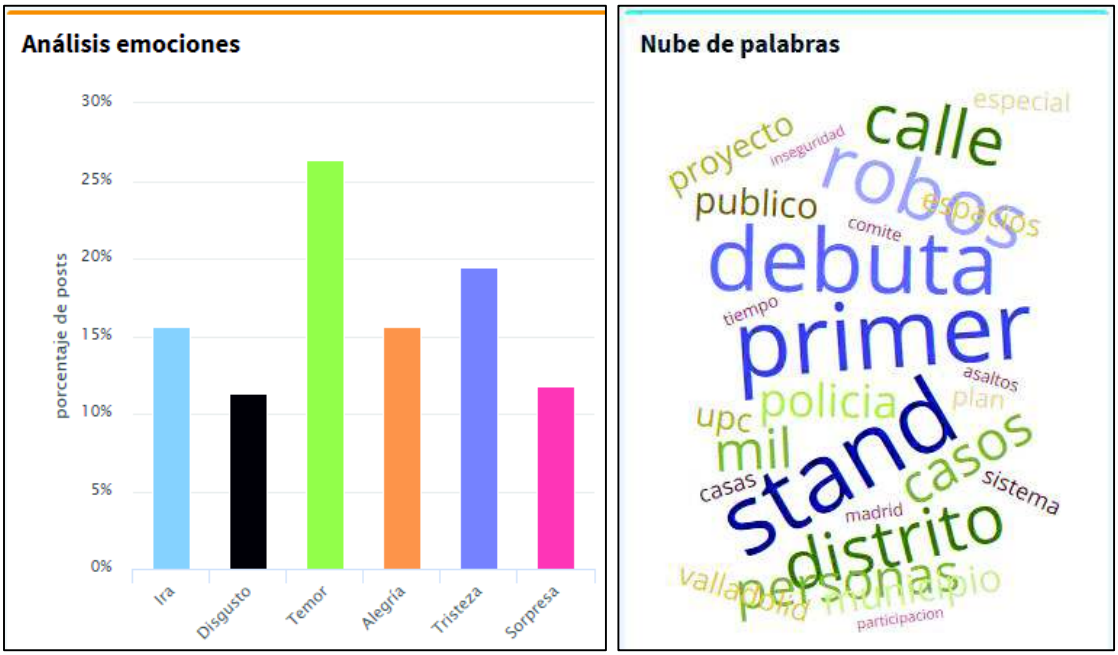
a) *b)*

Ilustración 55. La Floresta, análisis por temática ambiente. a) Histograma de emociones b) Nube de palabras

El medio ambiente posee un 73.44% de publicaciones positivas y un 4.69% negativas, como lo muestra la Ilustración 49. Gráfico de análisis de sentimientos por temática. Se demuestra un 35.26% de alegría, 16.05% de tristeza, un 13.16% de sorpresa, 12.37% de temor, un 11.84% de ira y 11.32% de desagrado, como lo indica la Ilustración 55.a.

Los principales tópicos fueron: huerta, sábado, botánica, espacio y arbolado, como se ve en la Ilustración 55.b. Donde se menciona a la “huerta”, que hace referencia a La Huerta, un vivero donde se pueden adquirir plantas para huertos urbanos; y “botánica”, que es un restaurante de comida orgánica, además se produce café artesanal.

3.1.2.7 Resultados por temática Inseguridad



a) *b)*

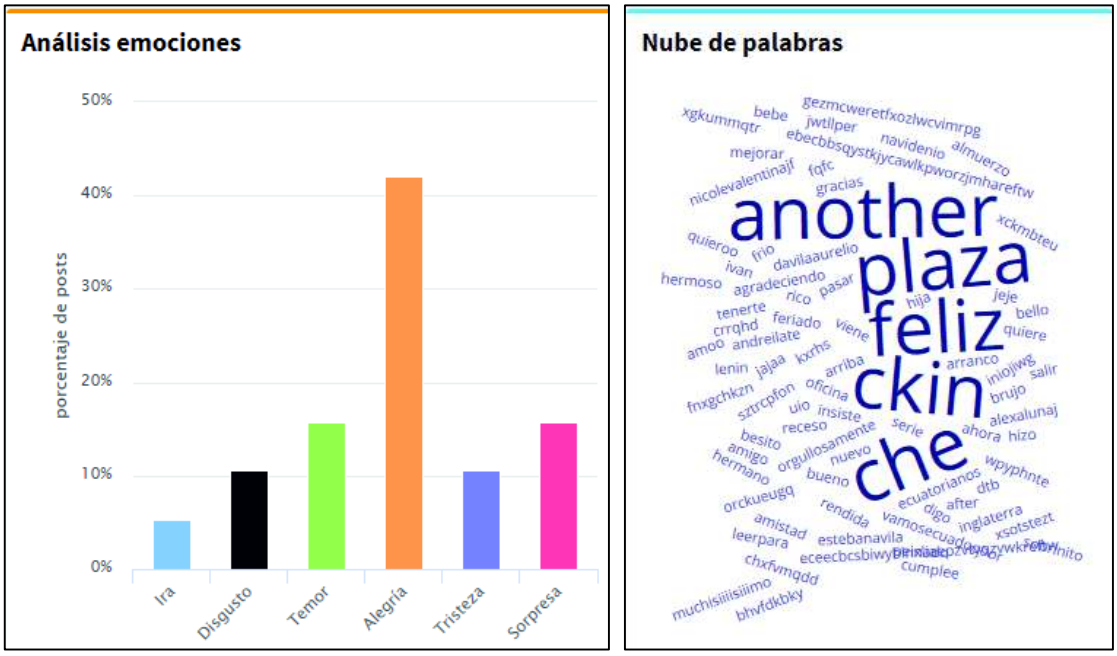
Ilustración 56. La Floresta, análisis por temática inseguridad. a) Histograma de emociones b) Nube de palabras

De inseguridad se tiene un 30% de publicaciones negativas, pero 60% de positivas, como se observa en la Ilustración 49. Gráfico de análisis de sentimientos por temática. La emoción detectada en mayor número de publicaciones fue el temor con un 26.38%, seguido de la tristeza con 19.42%, un 15.59% de ira, 15.59% de alegría, 11.75% de sorpresa y 11.27% de desagrado, como se puede ver en la Ilustración 56.a.

Los cinco tópicos más relevantes fueron: robos, casos, policía, primer, stand y distrito. Esto se ve en la Ilustración 56.b. Lo que junto con el análisis de emociones, muestran la preocupación de la gente con respecto a la delincuencia.

Un dato curioso de la nube de palabras es la aparición importante de las palabras: primer, stand y debuta; esto se debe a que en septiembre de 2017 se realizó en La Floresta un evento debut de comedia en vivo (en inglés *Stand Up Comedy*) llamado El Robo, éste fue muy publicitado en las redes sociales, describiendo al personaje principal de la obra como un ladrón y asesino. A pesar de tratarse de publicaciones que corresponden a la temática de Arte, el modelo los clasificó como un tema de inseguridad debido a la aparición de palabras referentes a este tema.

3.1.2.8 Resultados por temática Otro



a) *b)*

Ilustración 57. La Floresta, análisis por otras temáticas. a) Histograma de emociones b) Nube de palabras

Las publicaciones que no caben dentro de las temáticas mencionadas fueron asignadas a una categoría “otros”. Estas publicaciones muestran alegría en general, como lo indica la Ilustración 57.a.

En esta categoría se encontró los tópicos: plaza, feliz, y palabras extrañas, que posiblemente debido a ellas, los documentos no se lograron clasificar dentro de alguna de las temáticas identificadas. Estas palabras se muestran en la Ilustración 57.b. Un dato curioso es que muchas de las palabras que aparecen en esta categoría, no existen dentro del lenguaje español.

3.2 Discusión

Las dos capacidades de análisis más valorables del proyecto son la clasificación de texto y el análisis de sentimientos y emociones. Ambas tareas se llevaron a cabo con éxito y obteniendo resultados bastante satisfactorios. Sin embargo, en el caso de la clasificación de texto se tiene más control de la técnica que se está empleando. El hecho de poder realizar pruebas o transformar los datos de entrenamiento mediante funciones de limpieza, permite que el analista tenga la oportunidad de influir sobre la veracidad de los resultados que busca obtener. Por otro lado, el uso de librerías prefabricadas, como es el caso de Syuzhet, restringe el poder de mejorar los resultados que se obtienen de su aplicación. A partir de ambas tareas principales del sistema de minería de texto, las oportunidades de mejora están en aumentar la exactitud del modelo utilizado para la clasificación y buscar alternativas al uso de librerías prefabricadas para la determinación de los sentimientos y emociones.

La aplicación de modelos de análisis que consideran los textos como vectores multidimensionales o bolsas llenas de palabras, sin duda desaprovecha la característica semántica del texto. Esto daría la idea de que los enfoques usados en este proyecto no son del todo exactos. Por lo que un siguiente paso, para la evolución de la aplicación, sería probar métodos más complejos de análisis lingüístico, tanto los resultados como los costos en recursos de cómputo.

Un punto adicional de mejora es la detección automática del idioma. Esto puede ser necesario, debido a que muchas publicaciones, sobre todo en redes sociales, incluyen palabras o frases completas en inglés. Entre los datos se pueden tener publicaciones en otros idiomas de tal manera que, por ejemplo, para el análisis de sentimientos basado en diccionarios, este texto no podrá ser procesado.

4. CONCLUSIONES

- El presente proyecto, a pesar de no corresponder de manera específica a un caso de analítica de Big Data, se basó en una metodología pensada en el análisis de datos no estructurados y de grandes volúmenes. Las fases aplicadas se adaptaron perfectamente a las necesidades del proyecto.

- La adquisición de grandes volúmenes de datos no siempre es la mejor opción. La identificación de las fuentes y qué datos pueden proveernos juegan un papel fundamental en la construcción de la solución de analítica. Este fue el caso de Trip Advisor, del cual inicialmente se planeaba recolectar todos los comentarios publicados en los sitios turísticos ubicados en el barrio La Floresta. Esto con el fin de analizar los sentimientos de la gente y estimar el grado de aceptación que cada sitio posee. Mientras se llevaba a cabo el plan inicial, se halló que Trip Advisor exige la calificación del lugar a cada persona que publique un comentario sobre él. Trip Advisor recoge las calificaciones asignadas a cada sitio turístico y las promedia para establecer una calificación general. Esta calificación es ya el indicador que se deseaba estimar a través de análisis de sentimientos; lo cual hubiera significado un uso sin sentido de recursos.

- Un problema inevitable para todo sistema automático de recolección de datos web, es el mantenimiento del mismo. Esto en el sentido en que las fuentes de datos son específicamente sitios web, los cuales están en constante cambio. Por ejemplo, la renovación de la estructura de un sitio web, al que nuestro sistema accede a recoger datos mediante *web scraping*, dejaría obsoleta nuestra adquisición de datos de dicha fuente. Y no solamente ocurre con *web scraping*. A partir del 8 de abril del presente año, las políticas de Facebook cambiaron en varios aspectos. Uno de ellos, el acceso a sus datos a través de su API. A partir de esa fecha, es necesario solicitar directamente a Facebook un permiso especial para extraer datos de páginas públicas a través de su API. Este hecho dejó obsoleta la capacidad del sistema para recoger datos de esa fuente, hasta lograr obtener dicho permiso. Debido a que Facebook es la mayor fuente de datos del proyecto, como se puede observar en la Ilustración 32. Histograma de frecuencias absolutas: cantidad de documentos según la fuente., para futuros trabajos se propone llevar a cabo el proceso de solicitud de permisos de trabajo.

- Un requerimiento común en el ámbito de los sistemas de análisis, es la capacidad del sistema de recoger datos, procesarlos automáticamente y generar información, todo esto

en un instante de tiempo. El conocido análisis en tiempo real, en la práctica, no existe como tal. Toda tarea requiere de un tiempo para ser ejecutada con éxito, por más corto que éste sea. Esto deja solamente la posibilidad de construir sistemas que se aproximen al análisis en tiempo real. Además, el aprovechamiento de un sistema en tiempo real debe ir de la mano de la rapidez para captar cada cambio y tomar acciones basadas en los resultados del análisis. En el caso de minería de texto web acerca del barrio La Floresta, aproximadamente llegan dos registros cada día. Por lo que la implementación de sistemas de análisis cercanos al tiempo real en este caso, sería un gasto excesivo de recursos que no serían aprovechados. Esto se cumple debido a los requerimientos actuales del proyecto; sin embargo, para futuros trabajos, si se quisiera minar texto acerca de objetos sobre los que la afluencia de datos en la web es mucho mayor, se podría considerar el uso de sistemas de análisis que se acerquen al tiempo real.

- Para trabajos futuros, el sistema podría orientarse al procesamiento de datos acerca de entidades de mayor popularidad en la web. Por ejemplo, la minería de texto web acerca de las últimas investigaciones en el campo de la medicina. Esto serviría como una herramienta de actualización constante del conocimiento de los médicos del país. Para lograr esto, las tecnologías usadas en el proyecto, como CouchDB y R, están totalmente en la capacidad de escalar. Sin embargo, habría que realizar ajustes sobre el diseño de la arquitectura del sistema.

- La integración de datos, llevada a cabo en la fase de 2.5.Agregación y representación de datos, se realizó mediante un repositorio central de bases de datos de tipo no relacional. A diferencia de los tradicionales almacenes de datos, el repositorio utilizado no posee un modelo de datos interrelacionados. Para trabajos futuros, si el volumen de datos crece considerablemente, se debería considerar el uso de un indexador como *Elasticsearch*.

- Las tecnologías utilizadas, más concretamente R y CouchDB como principales herramientas, tienen dos características comunes. Ambas combinan facilidad de uso y potencia. R y su amplia variedad de librerías hacen posible un manejo y análisis de datos eficiente, a través de un lenguaje de programación de alto nivel y uso intuitivo. Al mismo tiempo, CouchDB ofrece fácil acceso a los datos a través de sus APIs, sin necesidad de configurar conectores u otros controladores que hacen que la compatibilidad entre tecnologías, sea un desgaste de recursos. Tanto CouchDB como R permiten al analista dedicarse al análisis y no a configurar ambientes complejos para desarrollo.

- El uso de métodos descriptivos y predictivos de minería de datos, dentro del mismo proyecto, funcionaron complementándose los unos con los otros. El análisis de sentimientos y el modelado de tópicos se llevaron a cabo mediante un enfoque descriptivo del texto. Por otro lado, de los métodos predictivos, las técnicas de clasificación o de aprendizaje supervisado se utilizaron para categorizar los textos en temáticas.

- Para seleccionar el modelo de clasificación a utilizar, se probaron tres de los más básicos modelos de aprendizaje supervisado, Naïve Bayes, árboles de decisión y máquinas de vectores de soporte. En las pruebas realizadas, se observó que Naïve Bayes, para nuestro conjunto de datos de entrenamiento y prueba, nunca superó el 30% de exactitud, aunque su tiempo de ejecución fue el segundo mejor. El modelo basado en árboles de decisión superó el 50% pero con el más largo tiempo de ejecución. Finalmente, el modelo basado en máquinas de vectores de soporte superó el 75% de exactitud, y con un tiempo de ejecución aproximadamente 60 veces menor que el tiempo de ejecución del modelo basado en árboles de decisión, y 30 veces menor al tiempo de ejecución del clasificador de Bayes. Se concluyó que, para nuestro conjunto de datos, el modelo de máquinas de vectores de soporte fue el más indicado.

5. REFERENCIAS

Aggarwall, C. (2015). *Data Mining: The Textbook* (Primera ed., Vol. I). Yorktown Heights, New York, U.S.A.: Springer. doi:10.1007/978-3-319-14142-8

Alpaydin, E. (2014). *Introduction to Machine Learning* (Tercera ed.). (T. Dietterich, Ed.) London, England: MIT Press. Recuperado el 12 de Abril de 2018, de https://mitpress.mit.edu/books/introduction-machine-learning-third-edition

Anderson, C., Lehnardt, J., & Slater, N. (2010). *CouchDB: The Definitive Guide* (Primera ed., Vol. I). Sebastopol, Rusia: O'Reilly. Recuperado el 28 de Abril de 2018, de http://guide.couchdb.org/

Ashish, K., & Avinash, P. (2016). *Mastering Text Mining with R* (Primera ed., Vol. I). Birmingham, U.K.: Packt Publishing. Recuperado el 12 de Abril de 2018, de https://www.packtpub.com/big-data-and-business-intelligence/mastering-text-mining-r

Berry, M., & Kogan, J. (2010). *Text Mining - Applications and Theory* (Primera ed., Vol. I). Chichester, U.K.: Wiley. Recuperado el 18 de Abril de 2018, de https://www.wiley.com/en-us/Text+Mining%3A+Applications+and+Theory-p-9780470749821

Brase, C., & Brase, C. (2013). *Understanding Basic Statistics* (Sexta ed., Vol. I). Boston, Massachusetts, U.S.A.: Cengage Learning. Recuperado el 20 de Abril de 2018, de http://www.nxtbook.com/nxtbooks/ngsp/basicstatistics6/

Buechler, S. (Mayo de 2007). *R Language Fundamentals.* Recuperado el Mayo de 25 de 2018, de University of Notre Dame - R course: https://www3.nd.edu/~steve/Rcourse/Lecture2v1.pdf

Cárcamo, L., Calva, D., Ronquillo, N., & Nesbet, F. (24 de Mayo de 2017). México, en la prensa chilena: análisis basado en minería de datos textuales en Twitter. *Revista Latina CS*, 897-914. Recuperado el 06 de Febrero de 2018, de http://www.revistalatinacs.org/072paper/1199/RLCS-paper1199.pdf

Chen, C. (25 de Julio de 2017). *We compared 7 travel-booking sites to show you what each is best at*. (Insider) Recuperado el 30 de Abril de 2018, de Insider Web Site: http://www.thisisinsider.com/expedia-tripadvisor-priceline-travel-booking-site-comparison/#bookingcom-1

Chen, M., Mao, S., & Liu, Y. (22 de Enero de 2014). Big Data: A Survey. *Mobile Networks and Applications, 19*, 171–209. doi:10.1007/s11036-013-0489-0

CRAN Project. (19 de Febrero de 2015). *Package 'RTextTools'.* (T. Jurka, Ed.) Recuperado el 16 de Mayo de 2018, de Comprehensive R Archive Network (CRAN) Project: https://cran.r-project.org/web/packages/RTextTools/RTextTools.pdf

CRAN Project. (29 de Agosto de 2016). *Package 'rvest'.* (H. Wickham, Ed.) Recuperado el 15 de Mayo de 2018, de Comprehensive R Archive Network (CRAN) Project: https://cran.r-project.org/web/packages/rvest/rvest.pdf

CRAN Project. (14 de Diciembre de 2017). *Package 'syuzhet'.* (M. Jockers, Ed.) Recuperado el 6 de Abril de 2018, de Comprehensive R Archive Network (CRAN) Project: https://cran.r-project.org/web/packages/syuzhet/syuzhet.pdf

CRAN Project. (6 de Diciembre de 2017). *Package 'tm'.* (I. Feinerer, Ed.) Recuperado el 15 de Mayo de 2018, de Comprehensive R Archive Network (CRAN) Project: https://cran.r-project.org/web/packages/tm/tm.pdf

De La Floresta. (Diciembre de 2016). *De La Floresta - Directorio*. Obtenido de delafloresta.com: http://www.delafloresta.com/

El Comercio. (10 de Mayo de 2018). *Resultados de búsqueda "La Floresta"*. Recuperado el 10 de Mayo de 2018, de Sitio web El Comercio: http://www.elcomercio.com/search/?query=la+floresta&_type=all&category=&publishedAt%5Bfrom%5D=&publishedAt%5Buntil%5D=2017-11-29&contentTypes%5B%5D=news&contentTypes%5B%5D=video&contentTypes%5B%5D=audio&contentTypes%5B%5D=photogallery&contentTypes%5B%5D=e

El Telégrafo. (15 de Mayo de 2018). *Resultados de búsqueda "La Floresta"*. Recuperado el 15 de Mayo de 2018, de Sitio web El Telégrafo: https://www.eltelegrafo.com.ec/busqueda?searchword=la%20floresta&ordering=newest&searchphrase=exact&limit=0

Erl, T., Khattak, W., & Buhler, P. (2016). *Big Data Fundamentals, Concepts, Drivers & Techniques* (Primera ed., Vol. I). New Jersey, United States: Prentice Hall. Recuperado el 29 de Noviembre de 2017, de https://www.pearson.com/us/higher-education/program/Erl-Big-Data-Fundamentals-Concepts-Drivers-Techniques/PGM328866.html

Facebook. (29 de Noviembre de 2017). *Resultado de búsqueda "La Floresta" (Páginas)*. Recuperado el 29 de Noviembre de 2017, de Sitio web Facebook: https://www.facebook.com/search/pages/?q=la%20floresta

Facebook. (2018). *API y SDK*. (Facebook) Recuperado el 20 de Mayo de 2018, de Facebook for developers: https://developers.facebook.com/docs/apis-and-sdks?locale=es_ES

Forbes. (30 de Abril de 2018). *Culture Trip: How A Former Shrink Found Millennial Travel Success*. (Forbes Europe) Recuperado el 7 de Marzo de 2018, de Forbes Web Site: https://www.forbes.com/sites/kittyknowles/2018/03/07/culture-trip-travel-app-for-millennials/#32baf38cf4ad

Ganis, M., & Kohirkar, A. (2016). *Social Media Analytics - Techniques and Insights for Extracting Business Value Out of Social Media* (Primera ed., Vol. I). Crawfordsville, Indiana, United States: IBM Press. Recuperado el 6 de Febrero de 2018, de http://www.pearson.com.au/products/D-G-Ganis-Kohirkar/Social-Media-Analytics-Techniques-and-Insights-for-Extracting-Business-Value-Out-of-Social-Media-VitalSource-eText/9780133892949?R=9780133892949

Gantz, J., & Reinsel, D. (December de 2012). *THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East*. Recuperado el Diciembre de 2016, de Dell EMC: https://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf

Gobierno Abierto Quito. (25 de Marzo de 2017). *Secretaría General de Planificación*. Recuperado el 29 de Marzo de 2017, de Población Económicamente Activa: http://datos.quito.gob.ec/datastreams/163/poblacion-economicamente-activa/

Gromski, P., Xu, Y., Turner, M., & Ellis, D. (Febrero de 2015). A tutorial review: Metabolomics and partial least squares-discriminant. *Analytica Chimica Acta*, 10-23. doi:10.1016/2015.02.012

Henrique, J. (29 de Enero de 2018). *GetOldTweets-python: Get Old Tweets Programatically*. Recuperado el 1 de Mayo de 2018, de GitHub Web Site: https://github.com/Jefferson-Henrique/GetOldTweets-python

Interactive Advertising Bureau (IAB) Ecuador. (13 de Noviembre de 2017). *Consumo Digital Ecuador 2017.* Recuperado el 30 de Abril de 2018, de IAB Ecuador: http://iabecuador.com/doc/EstudioDigital2017_IAB.pdf

Kloo, I. (Agosto de 2015). *Textmining: Clustering, Topic Modeling, and Classification.* Recuperado el 1 de Octubre de 2017, de Beskow Data Analytics Web Site: http://data-analytics.net/cep/Schedule_files/Textmining%20%20Clustering,%20Topic%20Modeling,%20and%20Classification.htm

Kwartler, T. (2017). *Text Mining in Practice with R* (Primera ed., Vol. I). Hoboken, New Jersey, U.S.A.: Wiley. Recuperado el 3 de Abril de 2018, de https://www.wiley.com/en-us/Text+Mining+in+Practice+with+R-p-9781119282013

Liu, B. (2015). *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions* (Primera ed., Vol. I). New York, U.S.A.: Cambridge University Press. doi:10.1017/CBO9781139084789

Loshin, D. (2002). *Enterprise Knowledge Management - The Data Quality Approach* (Primera ed., Vol. I). San Diego, California, U.S.A.: Academic Press. Recuperado el 2 de Marzo de 2018, de https://www.elsevier.com/books/enterprise-knowledge-management/loshin/978-0-12-455840-3

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook* (Primera ed., Vol. I). Beerseba, Israel: Springer. doi:10.1007/978-0-387-09823-4

Meyer, M., & Fisher, D. (2018). *Making Data Visual - A Practical Guide to Using Visualization for Insight* (Primera ed., Vol. I). Sebastopol, Rusia: O'Reilly. Recuperado el 26 de Abril de 2018, de http://shop.oreilly.com/product/0636920041320.do

Mohammad, S. M., & Turney, P. (Noviembre de 2017). *NRC Emotion Lexicon*. Recuperado el 6 de Abril de 2018, de NRC Word-Emotion Association Lexicon: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Moshenek, A. (24 de Abril de 2017). Entrevista telefónica con Representante del Colectivo "De La Floresta". Quito, Pichincha, Ecuador.

Munezero, M., Montero, C., & Mozgovoy, M. (2015). EmoTwitter – A Fine-Grained Visualization System for Identifying Enduring Sentiments in Tweets. En A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing. 2*, págs. 78-91. Cairo - Egipto: Springer. doi:10.1007/978-3-319-18117-2

Picot, J. (Septiembre de 2016). *An introduction to web crawler*. Recuperado el 22 de 02 de 2018, de Oncrawl: https://www.oncrawl.com/technical-seo/introduction-web-crawler/

Plutchik, R. (2003). *Emotions and Life: Perspectives From Psychology, Biology, and Evolution* (Primera ed., Vol. I). Washington, DC, U.S.A.: American Psychological Association. Obtenido de http://psycnet.apa.org/record/2003-04005-000

Posso, M. Á. (2011). *Proyectos, Tesis y Marco Lógico: Planes e Informes de Investigación* (Primera ed., Vol. I). Quito, Pichincha, Ecuador: Centro Editorial CEDISA. Recuperado el 26 de Abril de 2018

Quito Turismo. (Abril de 2013). *Empresa Pública Metropolitana de Gestión de Destino Turístico.* Recuperado el 28 de Febrero de 2017, de Quito en Cifras: http://www.quito-turismo.gob.ec/phocadownload/EstadisticasUIO/Quitoencifras/quito%20en%20cifras%202.pdf

Real Academia Española. (2014). *Diccionario de la lengua española (22.a ed.)*. Recuperado el 4 de Abril de 2018, de http://dle.rae.es

Runkler, T. A. (2016). *Data Analytics. Models and Algorithms for Intelligent Data Analysis* (Segunda ed., Vol. I). Munich, Germany: Springer. doi:10.1007/978-3-8348-2589-6

Sangaku Maths. (Marzo de 2018). *Frecuencia absoluta, relativa, acumulada y tablas estadísticas*. Recuperado el 20 de Abril de 2018, de Sangaku Maths Web Site:

https://www.sangakoo.com/es/temas/frecuencia-absoluta-relativa-acumulada-y-tablas-estadisticas

Secretaria de Territorio, Habitat y Vivienda - Municipio de Quito. (2013). *Secretaria de Territorio, Habitat y Vivienda - Municipio de Quito.* Recuperado el 6 de Febrero de 2017, de Parroquias de Distrito Metropolitano de Quito - Parroquia Urbana Mariscal Sucre: http://sthv.quito.gob.ec/images/indicadores/Barrios/mariscal.jpg

Servicio de Rentas Internas. (10 de Marzo de 2017). *Guía Básica Tributaria - RUC.* Recuperado el 28 de Febrero de 2017, de Base de datos del Registro Único de Contribuyentes de Personas Naturales y Sociedades: http://www.sri.gob.ec/DocumentosAlfrescoPortlet/descargar/fc895695-4548-4c5e-91a8-47ca67943739/PICHINCHA.zip

Shiny. (2013). *Building 'Shiny' Applications with R*. Recuperado el 25 de Mayo de 2018, de R Studio - Shiny Web Site: http://rstudio.github.io/shiny/tutorial/

Srivastava, A. N., & Mehran, S. (2009). *Text Mining: Classification, Clustering, and Applications* (Primera ed., Vol. I). Boca Raton, Florida, U.S.A.: CRC Press. Recuperado el 10 de Abril de 2018, de https://www.crcpress.com/Text-Mining-Classification-Clustering-and-Applications/Srivastava-Sahami/p/book/9781420059403

Strategyn. (Agosto de 2017). *Market Opportunity: Discover hidden growth opportunities*. Recuperado el 28 de Abril de 2018, de Strategyn Web Site: https://strategyn.com/outcome-driven-innovation-process/market-opportunity/

Sung-min, K., & Sung-min, K. (2016). Automated Discovery of Small Business Domain Knowledge Using Web Crawling and Data Mining. *2016 International Conference on Big Data and Smart Computing (BigComp)* (pág. 4). Hong Kong: IEEE. Recuperado el 10 de Enero de 2018, de ieee.org: http://ieeexplore.ieee.org/document/7425974

The Culture Trip. (2017). *TheCultureTrip.com*. Recuperado el 12 de Julio de 2017, de La Floresta Quito Ecuador: https://theculturetrip.com/?s=la%20floresta%20quito%20ecuador

The Culture Trip. (2018). *Quito Ecuador Searching Results*. Recuperado el 6 de Enero de 2018, de The Culture Trip: https://theculturetrip.com/?s=quito%20ecuador

Trip Advisor. (2017). *Search results for: La Floresta near Quito Ecuador*. Recuperado el 25 de Julio de 2017, de TripAdvisor.com: https://www.tripadvisor.com/Search?geo=294308&latitude=&longitude=&searchNearby=&pid=3826&redirect=&startTime=&uiOrigin=&q=la+floresta

Twitter Developers. (Noviembre de 2017). *Get tweets older than 10 days with twitter API*. (Twitter) Recuperado el 1 de Mayo de 2018, de Twitter Developers Forums: https://twittercommunity.com/t/get-tweets-older-than-10-days-with-twitter-api/97456

Wiley, J. (2015). *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data* (Primera ed., Vol. I). Indianapolis, Indiana, U.S.A.: EMC Education Services. Recuperado el 10 de Febrero de 2018, de https://www.wiley.com/en-ec/Data+Science+and+Big+Data+Analytics:+Discovering,+Analyzing,+Visualizing+and+Presenting+Data-p-9781118876138

Zhai, C., & Aggarwal, C. (2014). *Mining Text Data* (Primera ed., Vol. I). New York, U.S.A.: Springer. doi:10.1007/978-1-4614-3223-4

6. ANEXOS

6.1 Anexo 1. Registro de estandarización de campos

Campos \ Fuentes	Facebook		Twitter		TheCultureTrip		ElComercio		ElTelegrafo		TripAdvisor
	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	
id	id		id		id		id		id		
date	created_time	AAAA-MM-DD T h:m:s + ####	date	MM/DD/AAAA h:m:s	date	AAAA-MM-DD	fecha	dia DD/MM/AAAA	date_format	DD/MM/AAAA	
text	message		text		article_text		text		text		
author	faltante		name		author				author		
link	faltante		source		url		link		link		
likes	likes/summary/total_count		favorite_count								
location	faltante		location		locationName						
hashtags	faltante		hashtags								
user_mentions	faltante		user_mentions								
retweet_count	faltante		retweet_count								

Estandarización de campos

Campos \ Fuentes	Facebook		Twitter		TheCultureTrip		ElComercio		ElTelegrafo		TripAdvisor
	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	nombre de campo	formato	
id	id		id		id		id		id		
date	date	AAAA-MM-DD	date	AAAA-MM-DD	date	AAAA-MM-DD	date	AAAA-MM-DD	date	AAAA-MM-DD	
text	text		text		text		text		text		
author	faltante		name		author				author		
link	faltante		source		link				link		
likes	likes		likes				link				
location	faltante		location		location						
hashtags	faltante		hashtags								
user_mentions	faltante		user_mentions								
retweet_count	faltante		retweet_count								

6.2 Anexo 2. Función de traducción usando R y la API de Google Translate

```
# Archivo: translate.R
# Autor: Marlon Vargas
# Fecha: diciembre 2017

library(RJSONIO)
library(RCurl)

translate <- function(txtToTranslate){

  toTranslate <- txtToTranslate

  translated <- vector(mode="character", length=0)
  nvt <- ceiling(nchar(toTranslate)/5000)
  div <- floor(nchar(toTranslate)/ceiling(nchar(toTranslate)/5000))

  for (n in 1:(nvt+1)) {
    tmp_tslt <- vector(mode="character", length=0)
    tmp_json <- fromJSON(getURL(
      paste0(

   "https://translate.googleapis.com/translate_a/single?client=gtx&sl=en&tl=es&dt=t&q=",
        curlEscape(substr(toTranslate, (n-1)*div, (n*div)-1 ))
      )))

    for (m in 1:length(tmp_json[[1]])) {
      tmp_tslt <- paste(tmp_tslt, tmp_json[[1]][[m]][[1]], sep = " ",
   collapse = " ")
    }
    translated <- c(translated,tmp_tslt)
  }
  translated <- paste(translated, sep = " ", collapse = " ")

  return(translated)
}
```