*Sentime*: Análisis de sentimientos de los Colombianos sobre productos, servicios o personas a través de sus tweets.

Juan Sebastián Rodríguez Donado

Departamento de Ingenierías

Universidad del Norte

Barranquilla, Colombia

sjdonado@uninorte.edu.co

Tutor del proyecto: Ph.D Eduardo Enrique Zurek Varela

Departamento de Ingenierías

Universidad del Norte

Barranquilla, Colombia

ezurek@uninorte.edu.co

John Edison Fontalvo Pastorizo

Departamento de Ingenierías

Universidad del Norte

Barranquilla, Colombia

pastorizoj@uninorte.edu.co

Tutor del curso: Ph.D Wilson Nieto Bernal

Departamento de Ingenierías

Universidad del Norte

Barranquilla, Colombia

wnieto@uninorte.edu.co

***Abstract*— The information of people's opinion about different products, services or people is very important to make decisions in companies, but because of the high costs involved in surveying, this way of obtaining information is only affordable for large companies. With the rise of social networks such as Twitter, people are posting their opinion 24 hours a day about trending topics, but given the nature of social media this information is disorganized. In this paper we propose a solution that consists of a web platform capable of obtaining tweets at certain time intervals, analyzing the sentiment expressed in each tweet and providing the user with the sentiment classification about a specific search for each capital of the Departments of Colombia. The classification of the tweets for each search is achieved thanks to a previously trained neural network that has a 78% success rate.**

# Introducción

La percepción del consumidor sobre un producto, servicio o persona está relacionada directamente con la intención de consumo, por lo que, tanto empresas como figuras públicas en el marco del estudio de mercado elaboran encuestas para recolectar dicha información. Con el pasar de los años en Colombia han surgido diversas empresas que se encargan de optimizar estos procesos, esta es una alternativa factible si se cuenta con el capital suficiente, puesto que realizar encuestas a nivel nacional requiere de planeación y logística física con el objetivo de obtener la mayor cantidad de opiniones de los ciudadanos. Sin embargo, gracias a las redes sociales dicha opinión es depositada cada segundo, permitiendo omitir el paso de recolección física y por ende reduciendo costos.

Entre las redes sociales utilizadas en Colombia se encuentra Twitter, un servicio de microblogging en el cual por medio de textos cortos o Tweets las personas pueden compartir mensajes en texto plano (que puede estar acompañado de contenido multimedia), los cuales pueden ser opiniones acerca de temas variados. Esta red social ha evolucionado de gran manera, teniendo hoy en día bastante influencia sobre la sociedad y sobre la toma de decisiones en algunos aspectos[1].

En Colombia tanto personas comunes, como partidos políticos[2], canales de televisión, deportistas, personajes del gobierno, entre otros utilizan Twitter, y según MinTIC, se estima que la cifra de usuarios en 2019 de esta red es cerca de 6 millones en el país, superando a países como Francia y Alemania[3]. Por ende, disponer de esta gran cantidad de información nos brinda la oportunidad de que pueda ser procesada haciendo uso de técnicas de machine learning, que para el caso de los estudios de mercado es oportuno como caso base clasificar los sentimientos de los colombianos con respecto al producto o servicio ofrecido por la empresa, o incluso también para personas naturales como figuras públicas. A continuación, describiremos el desarrollo de esta aplicación la cual hemos llamado *Sentime*.

# Descripción del problema

Con el pasar de los años han surgido plataformas como servicio que ofrecen estudios de mercado, un ejemplo de estas es surveymonkey.com, la cual permite delimitar un público objetivo, realizar una encuesta basada en plantillas y analizar los resultados tanto en tiempo real como con informes personalizados. Además, antiguas compañías dedicadas a investigaciones de mercado como gfk.com se han visto obligadas a modificar sus procesos para tener mayor precisión y calidad en cuanto a la información que recopilan[4], esto se debe a que según Cristina Craciun “empresas muy grandes que estaban acostumbradas a luchar contra otras empresas grandes, ahora tienen un contexto competitivo mucho más dinámico”[4]. Aún así, estos servicios se basan en la recopilación de información a través de encuestas, lo que los obliga a ofrecer incentivos a los consumidores (que cumplan con las características definidas en el público objetivo) para responderlas, manteniendo un costo orientado a grandes empresas debido a que para el caso de surveymonkey este ronda entre los 3 millones COP (consultado en mayo del 2020).

# Justificación

Vivimos en una era de información, y las PYMES (pequeñas y medianas empresas) juegan un papel importante en el desarrollo tecnológico y económico del país. Según información del Ministerio de Comercio, Industria y Turismo, este tipo de empresas de menor tamaño representan 90% del total de las empresas del país, y no sólo eso, sino que crean 80% de los empleos nacionales y generan 50% del Producto Interno Bruto[5]. Sin embargo, cerca de la mitad de las pequeñas y medianas empresas del país se quiebra después del primer año y solo 20% sobrevive al tercero[6]. Por esta razón, proveer una herramienta gratuita que les permita obtener información con un margen de error aceptable, sobre la percepción (positiva o negativa) de sus posibles consumidores (Colombianos) dividida por departamentos, les brindaría información relevante en el marco de adaptación al mercado en su primer año de operación.

****

1. Mapa del problema

# Objetivos

## *Objetivo General*

Desarrollar una aplicación que permita clasificar los sentimientos sobre productos, servicios o personas a través de tweets utilizando técnicas de machine learning y visualización de datos.

## *Objetivos Específicos*

* Elaborar una revisión sistemática de la literatura asociada con el análisis de sentimientos y técnicas de machine learning.
* Implementar una red neuronal que permita clasificar los sentimientos de un Tweet en español en positivo, negativo o neutral.
* Elaborar un dataset de tweets en español ubicados geográficamente en Colombia usando Amazon Comprehend.
* Obtener tweets en tiempo real que estén ubicados geográficamente en Colombia a partir de una cadena de caracteres dada.
* Se tomarán las capitales de los 32 departamentos de Colombia para visualizar a través de un mapa de calor los resultados de la clasificación de los sentimientos.

# Metodología

Fase 1: Revisión bibliográfica

Esta etapa tiene como propósito elaborar la revisión sistemática de la literatura relacionada con técnicas de machine learning y análisis de sentimientos por medio de texto, a partir de la búsqueda de fuentes documentales en bases de datos IEEE-Xplore y Web of Science, que den cuenta de los temas previamente mencionados de tal forma que contribuya a elaborar el marco conceptual y a soportar el desarrollo de los diferentes elementos del trabajo de investigación.

Fase 2: Modelo de la solución

Esta fase tiene como fin desarrollar el modelo y diseño de la solución para implementar una plataforma web que permita analizar los sentimientos de los tweets de los ciudadanos colombianos, el modelo será entrenado con tweets en español ubicados en Colombia con el objetivo que pueda reconocer la jerga local, además dentro de su arquitectura contará con una capa LSTM que permitirá almacenar información referente al contexto de las oraciones.

Fase 3: Prototipado

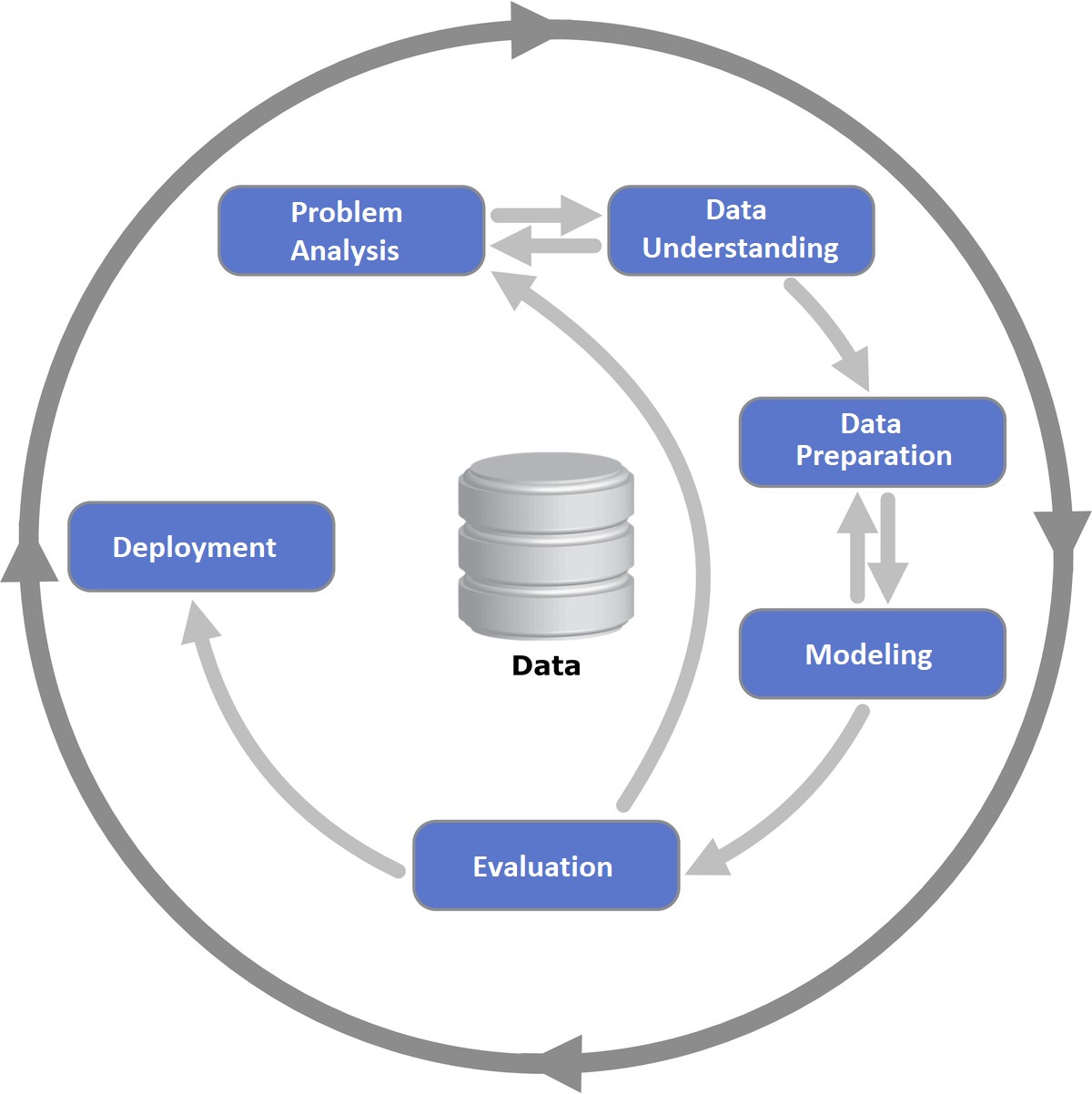
En esta fase se busca desarrollar el prototipo de la solución, en primera instancia se obtendrán tweets ubicados en Colombia utilizando la herramienta de código libre twint[7]. Estos tweets serán preprocesados eliminando caracteres diferentes al texto plano y posteriormente serán guardados en la nube (AWS S3), utilizando el modelo pre entrenado en español DetectSentiment de Amazon Comprehend se clasificarán los tweets en las siguientes categorías: “Mezclado”, “Positivo”, “Negativo” y “Neutral”. Estos tweets a excepción de la categoría mezclado serán usados para entrenar la red neuronal.

La aplicación permitirá que el usuario ingrese con un usuario y contraseña dado por el equipo de desarrolladores. Dentro de la plataforma este podrá ingresar una cadena de texto y un intervalo de tiempo de 2, 8 o 24 horas, al hacer click en buscar se desplegará en tiempo real el número de tweets obtenidos por cada capital de los departamentos de Colombia. A medida que se terminen de obtener los tweets para un departamento, estos serán clasificados usando la red entrenada previamente, el mapa de calor se actualizará dinámicamente cada vez que se obtenga la clasificación para un departamento. Finalmente, cada búsqueda del usuario será guardada con el fin de que pueda acceder posteriormente a ella, además todos los usuarios podrán acceder a todas las búsquedas realizadas en la plataforma.

Fase 4: Validación

Para validar los resultados se utilizará “validación cruzada” en el proceso de entrenamiento de la red con el objetivo de obtener un modelo con bajo porcentaje de error.

Fase 5: Metodología de desarrollo



1. Diagrama Metodología CRISP-DM

Para la presente investigación se hizo uso de la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), modelo estándar en el que

La metodología CRISP-DM consta de seis fases:

* Problem Analysis: En esta fase se analizan los objetivos bajo una perspectiva concreta, en este caso desde la perspectiva de análisis de sentimientos.
* Data Understanding: Durante esta fase se recolectan los datos y se analizan de cerca.
* Data Preparation: Una vez recogidos los datos, en esta fase, se preparan y se limpian los mismos.
* Modeling: En esta fase se detectan los patrones que puedan tener los datos que fueron preparados previamente
* Evaluation: Durante esta fase se determina si los resultados son de verdad útiles para la investigación y se evalúan los procesos efectuados para mejorar su uso práctico.
* Deployment: En la fase final se efectúa la difusión de la información de tal manera que la información ayude a tomar una decisión al usuario.

# Marco Conceptual

El análisis de sentimientos o minería de opinión es el proceso de determinar el tono emocional que hay detrás de una serie de palabras, y se utiliza para entender las actitudes, opiniones y emociones expresadas en una mención online.

Los primeros trabajos en esta área incluyen Turney[8] y Pang[9], los cuales aplicaron diferentes métodos para detectar la polaridad de críticas de productos y de películas respectivamente.

Gran cantidad de trabajos se han hecho respecto al área de análisis de sentimientos o minería de opinión. Entre los aportes más significativos se encuentran:

Soni, A. K. [10] usando sistemas de clasificación elaboró un sistema para filtrar información importante de twitter para luego analizar los sentimientos que esos tweets pretenden representar.

Kaladevi, P., Thyagarajah, K. [11] propusieron una metodología ICNN-LSTM-DNN (Integrated Convolutional Neural Network and Long Short Term Memory Recurrent Neural Network-based Deep Neural Networks-based) para la minería de opinión en tiempo real.

Zhang, J., Li, Y., Tian, J., & Li, T. [12] propusieron un modelo híbrido que combina las características y ventajas de dos modelos neurales tradicionales: LSTM (Long Short Term Memory) y CNN (Convolutional Neural Network) para mejorar el accuracy de la clasificación de texto a comparación de los dos modelos mencionados.

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. [13] en su investigación plantean una mejora del modelo de word representation generando vectores de palabras ponderadas, y sus resultados los comparan con métodos como RNN, CNN, LSTM, y NB, mostrando que el modelo propuesto tiene mejor precisión que los mencionados previamente.

Ceron-Guzman, J. A., & Leon-Guzman, E. [14] usaron clasificación supervisada para desarrollar el sistema de análisis de sentimientos en aras de determinar los sentimientos de las personas en las elecciones de Colombia en 2014.

Chen, B., Huang, Q., Chen, Y., Cheng, L., & Chen, R. [15] presentaron un modelo híbrido basado en modelos CNN (Convolutional Neural Network) y LSTM (Long Short Term Memory), evidenciando resultados con mejor accuracy qué máquinas de vectores de soporte (SVM), CNN, LSTM y CNN-LSTM existentes.

Paredes-Valverde, M. A., Colomo-Palacios, R., Salas-Zárate, M. del P., & Valencia-García, R. [16] exponen una solución basada en CNN y word2vec para permitirle a las compañías y organizaciones detectar oportunidades para mejorar la calidad de sus productos o servicios a través de análisis de sentimientos de tweets.

Suresh, H., & Gladston Raj S. [17] proponen un modelo de fuzzy clustering para analizar tweets buscando analizar los sentimientos hacia una marca en particular, para luego comparar los resultados con otros métodos de clustering existentes.

Parveen, H., & Pandey, S. [18] usaron el algoritmo de Naive Bayes para hacer análisis de sentimientos en Twitter, entrenando su modelo con un dataset clasificado de opiniones acerca de películas.

Kusrini, & Mashuri, M. [19] implementaron la técnica de procesamiento de lenguaje natural (NLP) para el análisis de sentimientos de tweets, usando léxicos y polaridad de multiplicación en vez de machine learning.

Con respecto al tema de mapeado de calor, hay un aporte bastante significativo, que es el de Huacon, C. F., & Pelegrin, L. [20] en el que desarrollaron SURV, un sistema de visualización de ciudades utilizando las librerías de Javascript Mapbox.js y CARTO.js para crear mapas interactivos y dinámicos.

Web scraping es una técnica utilizada para extraer información de sitios web. En esta área hay aportes como:

Diouf, R., Sarr, E. N., Sall, O., Birregah, B., Bousso, M., & Mbaye, S. N. [21] en el cual exponen los distintos métodos actuales de obtener información de websites haciendo web scraping.

La revisión bibliográfica de la literatura se realizó en varias fases, buscando términos claves acerca del tema en el que se está trabajando, luego filtrando todos los resultados para obtener los más recientes (artículos e investigaciones de 2016 al presente 2020), posterior a esto se leen los abstract de cada artículo para hacer un overview del contenido e inferir si potencialmente puede contribuir al proyecto, y por último se lee en profundidad cada artículo para determinar con certeza si es útil para usarlo como referencia en el proyecto.

Fase 0:

Fuentes de información:

* IEEE-Xplore
* Web of Science

Conceptos Core:

* Web heatmap data visualization
* Twitter sentiment analysis
* CNN-LSTM neural networks
* Web scraping

Fase 1: Búsqueda por conceptos (Query search - all metadata)

|  |  |  |
| --- | --- | --- |
| **Concepto/Fuente** | **IEEE** | **Web of Science** |
| Web heatmap data visualization | 21 | 41 |
| Twitter sentiment analysis | 1384 | 2085 |
| CNN-LSTM neural networks | 229 | 130 |
| Twitter big data | 136 | 1245 |
| **Total** | **1770** | **2590** |

Fase 2: Filtro por año (2016-2020) - H5 rule

|  |  |  |
| --- | --- | --- |
| **Concepto/Fuente** | **IEEE** | **Web of Science** |
| Web heatmap data visualization | 13 | 25 |
| Twitter sentiment analysis | 1006 | 1586 |
| CNN-LSTM neural networks | 229 | 129 |
| Twitter big data | 105 | 967 |
| **Total** | **1353** | **1974** |

Fase 3: Lectura del abstract

|  |  |  |
| --- | --- | --- |
| **Concepto/Fuente** | **IEEE** | **Web of Science** |
| Web heatmap data visualization | 5 | 12 |
| Twitter sentiment analysis | 27 | 41 |
| CNN-LSTM neural networks | 19 | 14 |
| Twitter big data | 23 | 25 |
| **Total** | **74** | **92** |

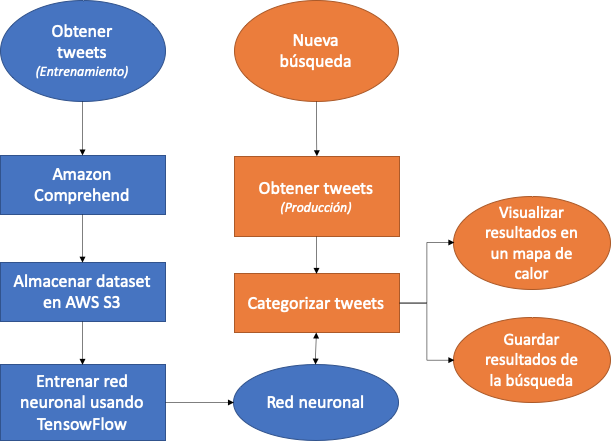
Fase 4: Lectura en profundidad

|  |  |  |
| --- | --- | --- |
| **Concepto/Fuente** | **IEEE** | **Web of Science** |
| Web heatmap data visualization | 1 | 0 |
| Twitter sentiment analysis | 4 | 2 |
| CNN-LSTM neural networks | 1 | 1 |
| Twitter big data | 1 | 2 |
| **Total** | **7** | **5** |

Artículos finales

|  |  |  |  |
| --- | --- | --- | --- |
| Título | Autores | Año | Keywords |
| Influence of media on the political  conversation on Twitter: Activity, popularity,  and authority in the digital debate in Spain | Casero-Ripollés, A. | 2020 | Political communication; Influence; Twitter; Media; Newspapers; Television;  Radio stations; News agencies; Political conversation; Interactions; Big data |
| Political Communication and Social Networks. Twitter: Comparative Analysis of the Presidencies of the Republic of Latin America. | Puertas-Hidalgo, R., Carpio-Jimenez, L., & Suing, A. | 2019 | internet; social network; Twitter; politics communication. |
| Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. | Peter Turney | 2002 | Classification; semantic orientation; |
| Thumbs up? Sentiment classification using machine learning techniques | Pang, B., Lee, L., Vaithyanathan, S. | 2002 | Sentiment classification; |
| Multi-lingual sentiment analysis of Twitter data by using classification algorithms | Soni, A. K. | 2017 | NLTK, Maximum Entropy, Naïve Bayes, N-gram Model, language independent, , text categorization and Twitter API |
| Integrated CNN- and LSTM-DNN-based sentiment analysis over big social data for opinion mining | Kaladevi, P., & Thyagarajah, K. | 2019 | Opinion mining; convolutional neural network; long short-term memory; social media posts; twitter messages |
| LSTM-CNN Hybrid Model for Text Classification | Zhang, J., Li, Y., Tian, J., & Li, T. | 2018 | Text Classification; Nature Language Processing; LSTM; Long Text Sequences; CNN |
| Sentiment Analysis of Comment Texts Based on BiLSTM | Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. | 2019 | Sentiment analysis, artificial intelligence, social network, weighted word vectors, BiLSTM |
| A Sentiment Analysis System of Spanish Tweets and Its Application in Colombia 2014 PresidentialElection | Ceron-Guzman, J. A., & Leon-Guzman, E. | 2016 | Supervised classification; sentiment analysis |
| Deep Neural Networks for Multi-class Sentiment Classification | Chen, B., Huang, Q., Chen, Y., Cheng, L., & Chen, R. | 2018 | Sentiment representation, CNN, LSTM, one-versus-restraining mechanism, sentiment classification |
| Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach | Paredes-Valverde, M. A., Colomo-Palacios, R., Salas-Zárate, M. del P., & Valencia-García, R. | 2017 | word2vec; CNN; sentiment classifier |
| An Unsupervised Fuzzy Clustering Method for Twitter Sentiment Analysis | Suresh, H., & Gladston Raj S. | 2016 | Sentiment Analysis (SA); partitioning clustering techniques; Expectation Maximization (EM); Simple K-Means. |
| Sentiment analysis on Twitter Data-set using Naive Bayes algorithm | Parveen, H., & Pandey, S. | 2016 | Twitter; Sentiment analysis; Algorithm design and analysis; Data mining; Computer architecture; Prediction algorithms |
| Sentiment Analysis In Twitter Using Lexicon Based and Polarity Multiplication | Kusrini, & Mashuri, M. | 2019 | twitter; sentiment analysis; NLP; lexicon |
| SURV: A System for Massive Urban Data Visualization | Huacon, C. F., & Pelegrin, L. | 2017 | smart; data; urban; visualization; heatmap; animation; Smart Cities |
| Web Scraping: State-of-the-Art and Areas of Application | Diouf, R., Sarr, E. N., Sall, O., Birregah, B., Bousso, M., & Mbaye, S. N. | 2019 | Web-Scraping; Data Collection; Web Data Extraction |

# Modelo de la solución



1. Modelo de la solución

Para el primer flujo (azul) en la etapa de obtener tweets se tomarán los que hayan sido publicados en los últimos 60 días y con ubicación en Colombia, esto con el objetivo de generar un dataset para entrenar la red neuronal.

Para el segundo flujo (naranja), el usuario después de ingresar con sus credenciales a la plataforma, podrá realizar búsquedas introduciendo texto plano y un intervalo de tiempo de 2, 8 o 24 horas, se obtendrán la mayor cantidad de tweets que cumplan con estas condiciones y que estén ubicados geográficamente en Colombia (con un límite de 80 por departamento). Esta tarea se realizará se distribuirá entre 6 hilos en paralelo, y se enviará información al usuario información del estado en tiempo real. Al finalizar la obtención y categorización de los tweets se mostrará el mapa de calor y se guardarán los resultados de la búsqueda, los cuales pueden ser consultados por el usuario dentro de la misma aplicación.

Con respecto a la arquitectura de la solución [fig. 4] se tiene como primera capa la máquina virtual de Docker, utilizando Docker Compose emulamos el mismo ambiente de desarrollo que el servidor de producción y con esto sólo necesitamos realizar la configuración para la integración de los servicios una sóla vez. El contenedor base (amarillo) se deriva de la imagen *python:3.6.9*, la cual tiene preinstalado y configurado el lenguaje python 3 y así mismo las dependencias necesarias para utilizar pytorch[22]. Este último, es el framework de código abierto que utilizaremos para entrenar, guardar y obtener información de nuestra red neuronal. Sobre el contenedor base también se encuentra el micro-framework Flask[23] que es el encargado de levantar y gestionar el servidor web (renderizado del aplicativo tipo server-side), además este hace uso de las librerías Twint (scraper para obtener tweets), Socket.IO[24] (conexiones en tiempo real), Google Maps[25] que gracias a su herramienta HeatmapLayer se puede visualizar el mapa de calor y Chart.js[26] para los diagramas de pie y barras.

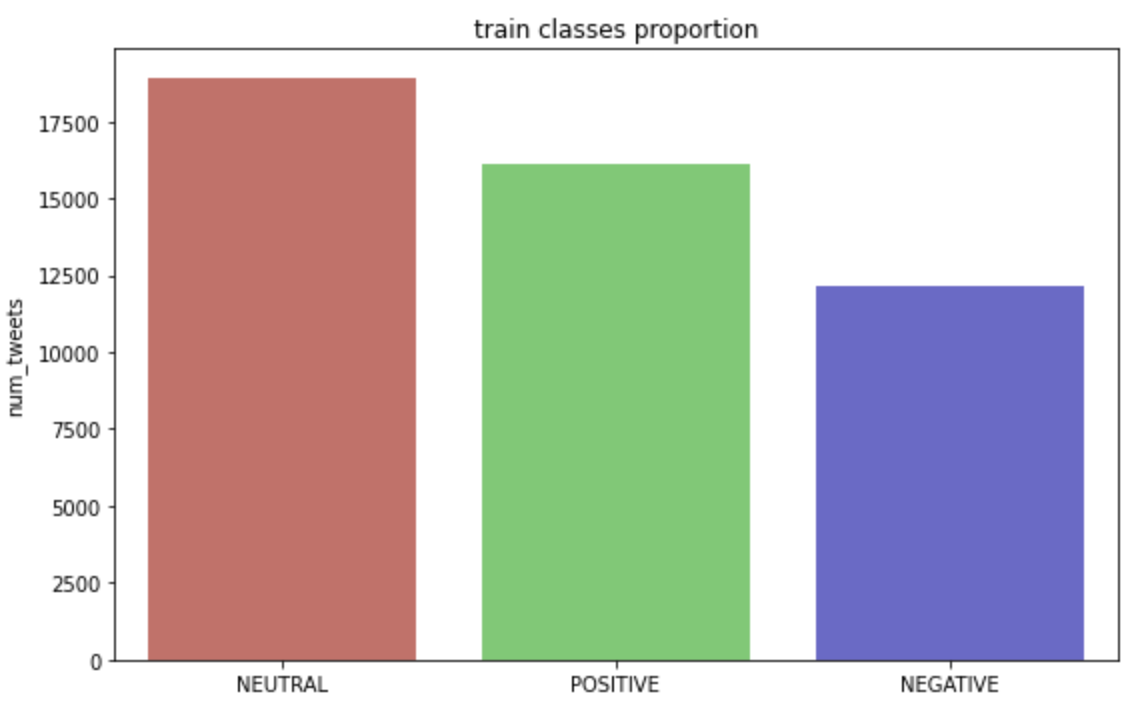


1. Arquitectura lógica de la solución

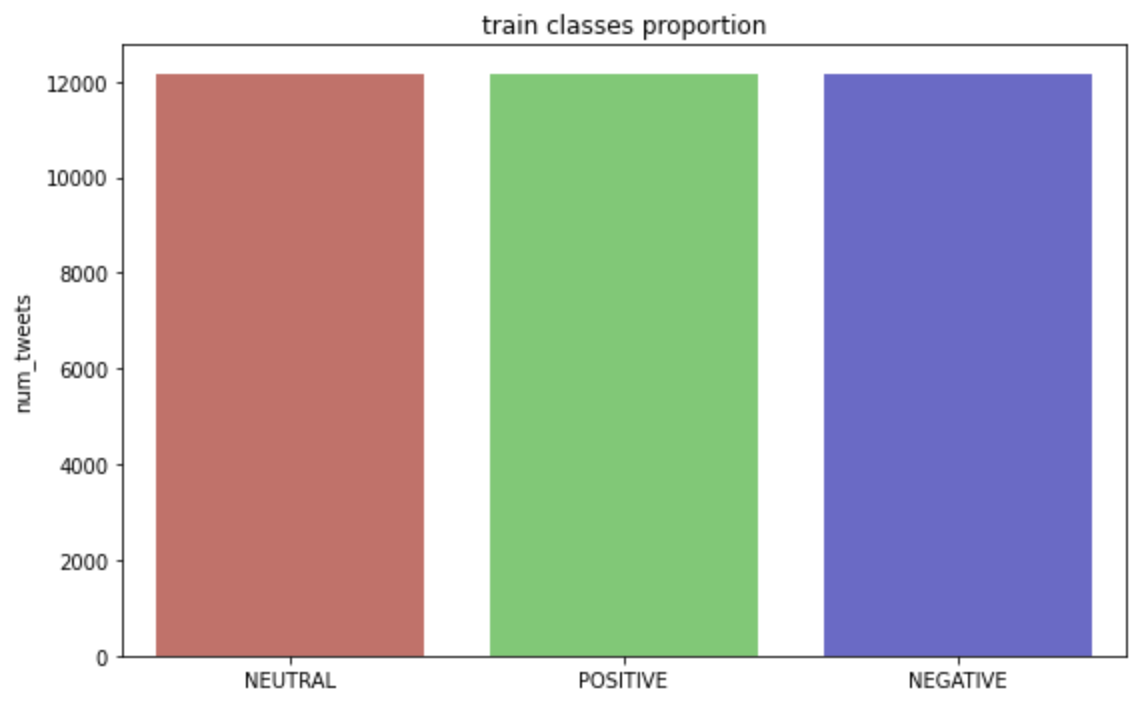
Por último, desde el contenedor base se tienen conexiones con tres servicios, dos dentro de la misma red provista por Docker (cada uno con su propio contenedor), y uno externo con la red de Amazon Web Services (esta sólo se realizará en la etapa de entrenamiento). Los pertenecientes a la red de Docker son los siguientes: Redis[27] (base de datos en memoria) que nos permite guardar información relacionada con el manejo de las sesiones y PostgreSQL[28] (base de datos en disco) en la cual se guardarán las búsquedas de los usuarios. A pesar que no se tiene una conexión automatizada con Amazon Comprehend, esta es necesaria en la etapa de entrenamiento, manualmente se crea un trabajo y se cargan los tweets obtenidos a un bucket en AWS S3, desde ahí son llamados por el REST API de Comprehend para ser procesados, una vez finalizado el trabajo se pueden descargar los resultados contenidos en un archivo alojado en el mismo bucket.

# Resultados

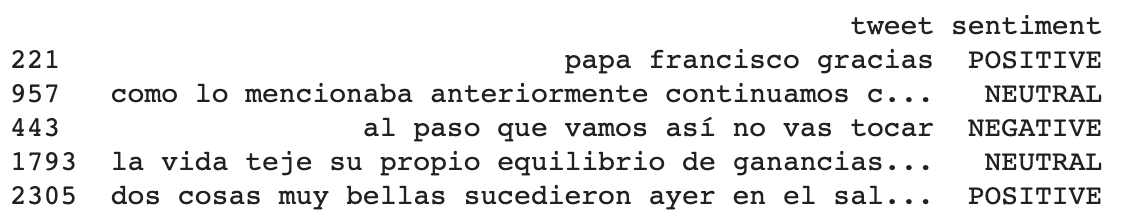
Para la recolección de los tweets a procesar en Amazon Comprehend se realizaron dos ejecuciones del scraper, la primera el 27/04/2020 con la cual se obtuvieron 30633 tweets, y la segunda el 24/05/2020 con un resultado de 53366 tweets. Mezclando estos dos datasets y eliminando los repetidos se procesaron en Amazon Comprehend 83998 tweets que fueron divididos en 67198 para entrenamiento, 12599 para validación y 4201 para pruebas [fig. 5]. La estructura del dataset está definida por dos columnas, una para el tweet en texto plano y otra para la categoría a la que pertenece [fig. 7].



1. Distribución de tweets por clases para el dataset de entrenamiento.

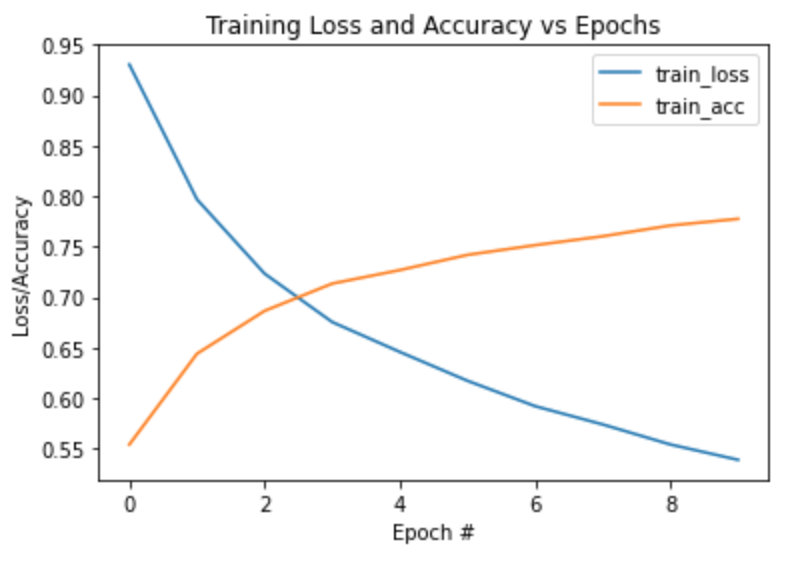


1. Distribución de tweets por clases para el dataset de entrenamiento balanceado



1. 5 filas seleccionadas al azar del dataset de pruebas

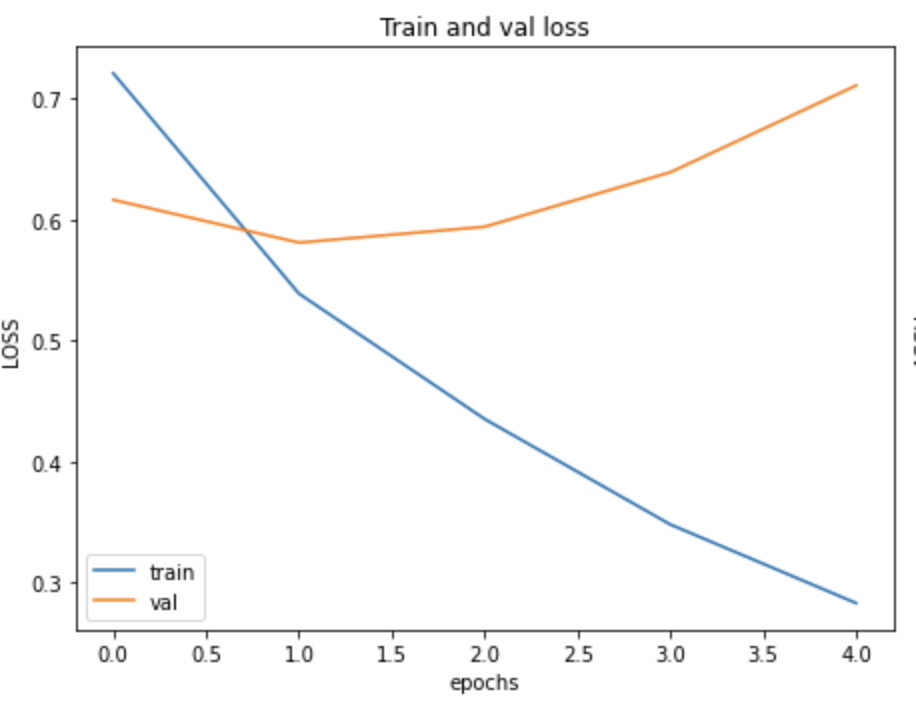
Para la construcción de la red neuronal en primera instancia se utilizó el embedding *es\_core\_news\_md* de la librería open source *Spacy*[29] y el tokenizer *sentencizer* provisto por la misma. La arquitectura por capas estaba constituida por: La capa del embedding para ser re entrenado, una capa LSTM bidireccional y finalmente una fully connected. El resultado obtenido fue un 72% de precisión para la data de entrenamiento y 63% de precisión para el dataset de pruebas (tweets nunca vistos antes), por lo que se optó por balancear el dataset de entrenamiento [fig. 6] (técnica utilizada por [16]) y ajustar parámetros como la tasa de aprendizaje, obteniendo finalmente un 77% para data de entrenamiento y 65% para test [fig. 8].



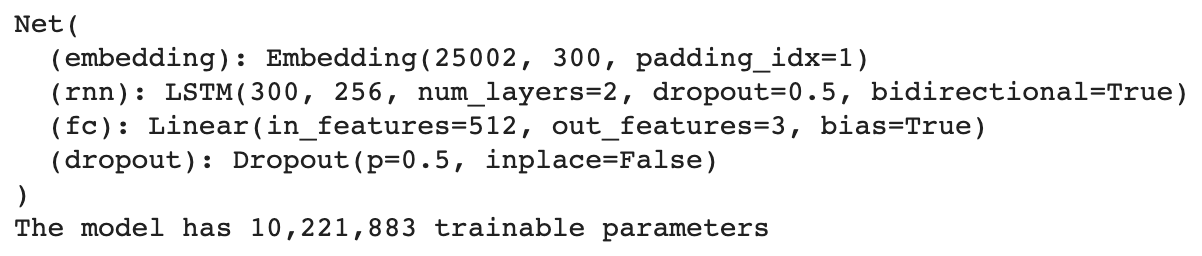
1. Precisión con respecto al número de iteraciones en el entrenamiento de la primera versión de la red neuronal.

Siguiendo la estructura propuesta en [13] reemplazamos el embedding provisto por Spacy por SBWC[30], el cual es una implementación de Word2Vect con Skygram usando 1.4 billones de palabras extraídas de Wikipedia. El dataset sigue siendo preprocesado usando Spacy pero el tokenizer es reemplazado por vocab de torchtext, la estructura de la red no es modificada por lo que el modelo ahora cuenta con 10’200.833 parámetros entrenables [fig. 10].

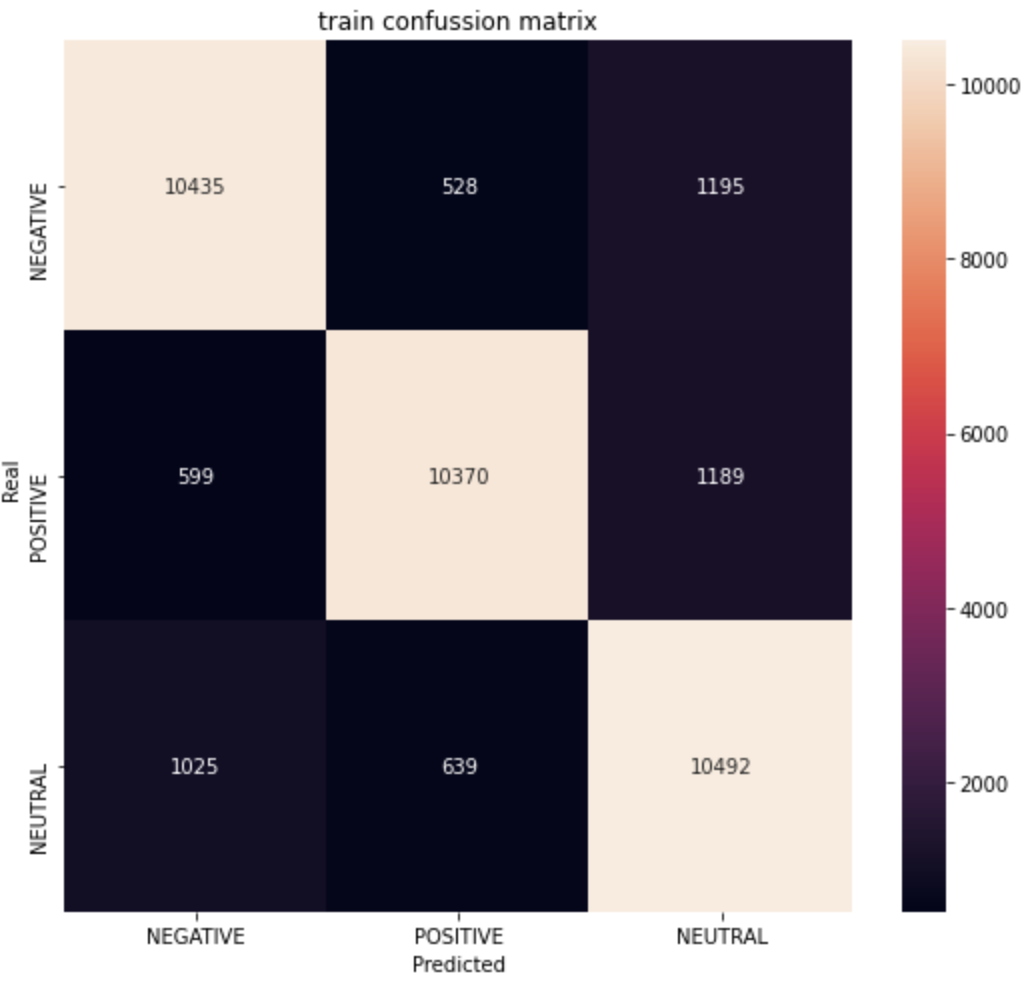
Para el entrenamiento de esta segunda versión, se redujeron el número de iteraciones [fig. 9] dado que no se modificó la tasa de aprendizaje y se tienen más parámetros entrenables, el tiempo de entrenamiento cambió de un promedio de 227 segundos por iteración a 6 minutos. Como resultado se obtuvo un 85.8% de precisión para la data de entrenamiento [fig .12] y un 78% para test [fig. 13].



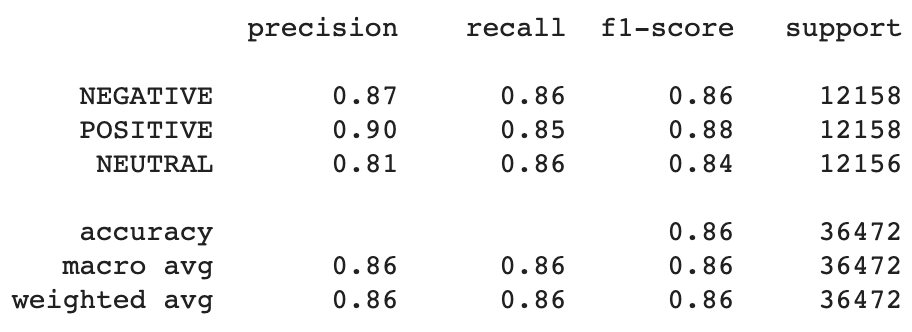
1. Precisión con respecto al número de iteraciones en el entrenamiento de la segunda red neuronal



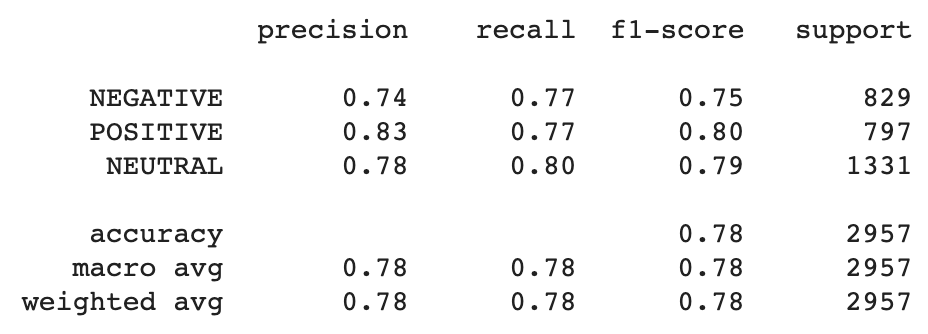
1. Arquitectura de la red



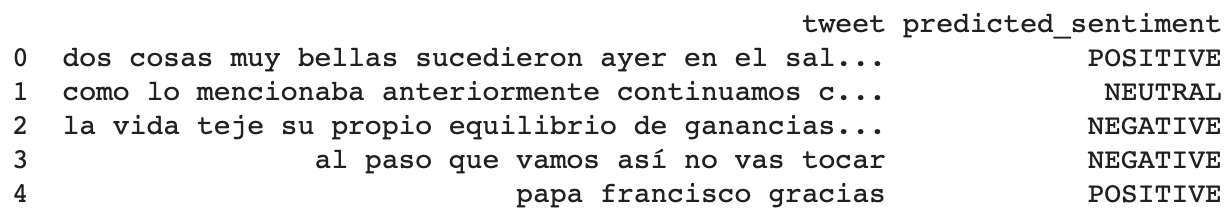
1. Matriz de confusión para el dataset de entrenamiento.



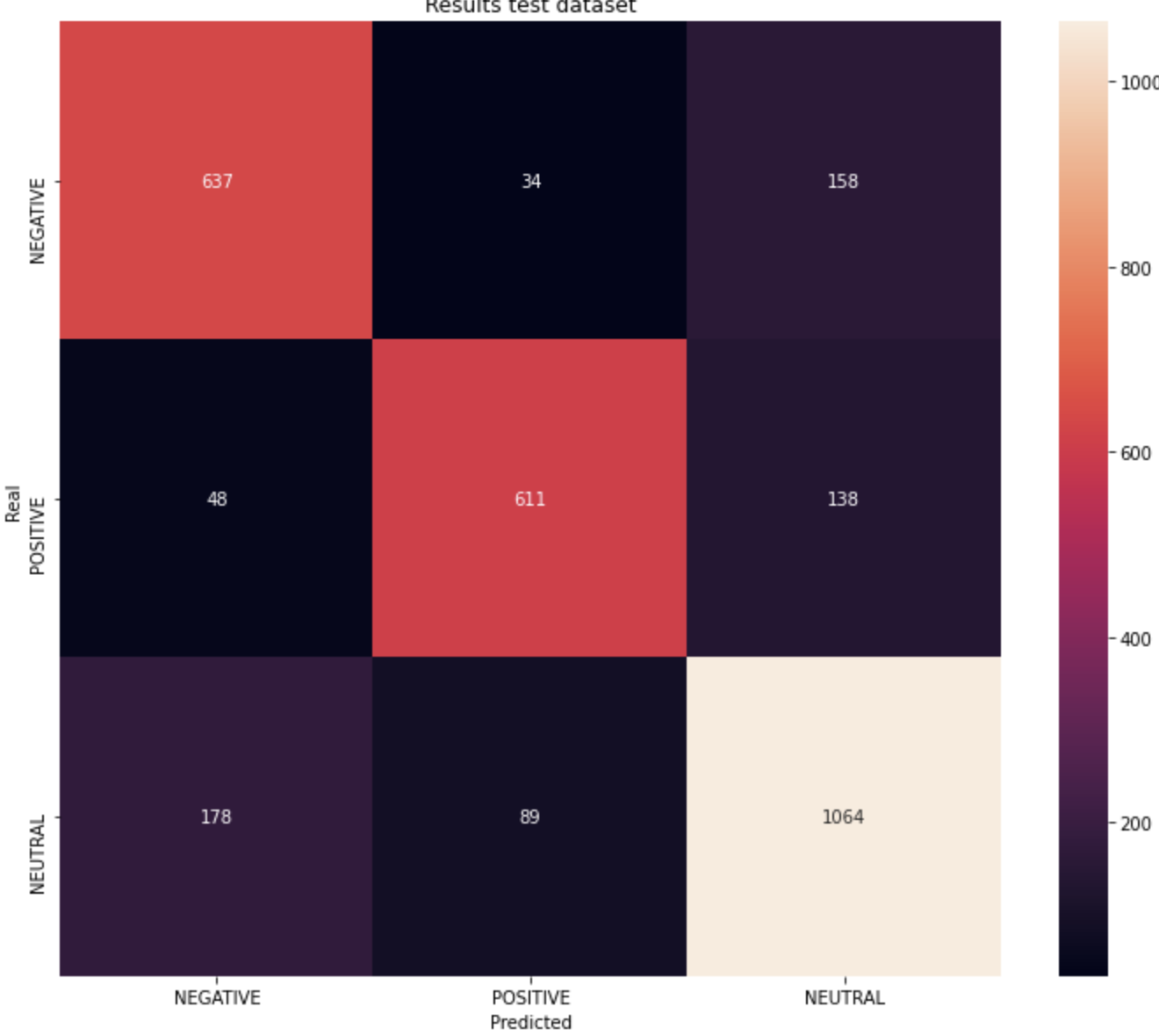
1. Resultados precisión para el dataset de entrenamiento.



1. Resultados precisión para el dataset de pruebas.



1. Predicciones obtenidas para las filas seleccionadas en la fig 7.



1. Matriz de confusión para las predicciones evaluadas en el dataset de pruebas (tweets nunca antes vistos por la red)

Dada la gran cantidad de información a renderizar, se tomó la decisión de simplificar la tarea del servidor al sólo actuar como API para realizar las búsquedas de los tweets e interactuar con la base de datos, por lo tanto, la información de los resultados son renderizados usando la librería open-source React[31] en el lado del cliente. Para que estas partes puedan comunicarse entre sí configuramos un servidor Nginx[32] como proxy reverso, teniendo como únicos puertos de salida el 80 y 433 desde la instancia en AWS EC2.

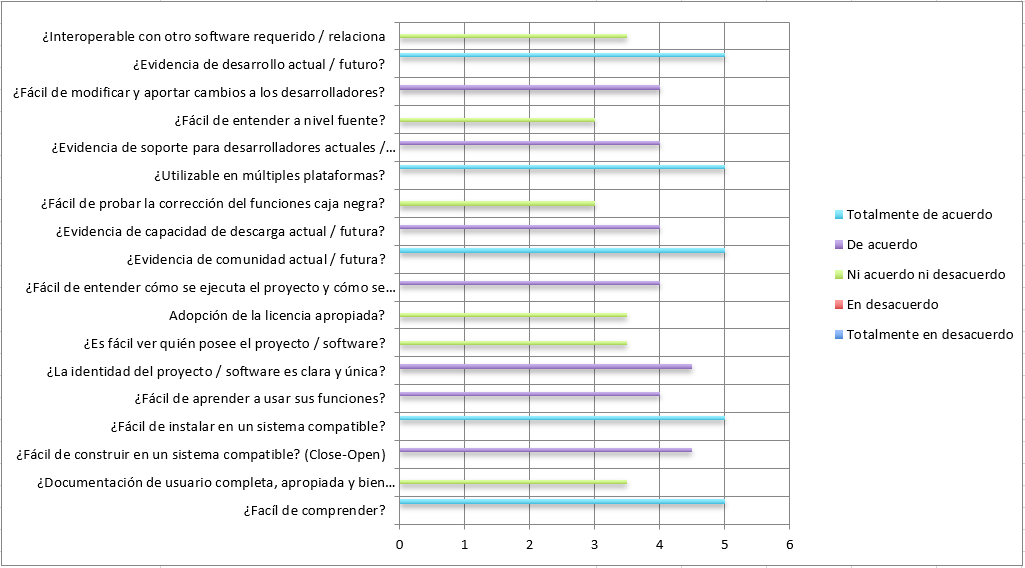
Para el despliegue de la plataforma se optó por AWS EC2, se adquirió el dominio sentime.com.co y se generó un certificado SSL haciendo uso de Certbot[33], la instancia utilizada es de tipo *t2.medium* (4GiB ram y 2 vCPUs) ya que la carga de modelo en memoria sobrepasaba los límites otorgados por la opción gratuita *t2.micro*. Dicha instancia cuenta con una dirección IPv4 única pública 3.230.95.119 y configurada con al dominio sentime.com.co a través del gestor de DNS AWS Route 53.

Los modelos fueron entrenados en Google Colab, dado que esta herramienta gratuita nos ofrece recursos limitados por sesiones correspondientes a 12.72GB RAM y 11439 MB GPU, suficientes para el entrenamiento de nuestra red. El scraper fue ejecutado en entornos locales controlados durante aproximadamente 18 horas, posterior a esto los tweets fueron preprocesados (remover emojis, links, hashtags, símbolos, etc haciendo uso de expresiones regulares), subidos a AWS S3, procesados por AWS Comprehend, descargados y mezclados para generar el dataset final, este proceso también fue realizado localmente a través de scripts diseñados en python.

Para realizar actualizaciones en el modelo se tiene que ejecutar el notebook y luego ser descargar los resultados en la carpeta de recursos del repositorio alojado en Github, estos pasos están detallados en el README del mismo. La conexión al servidor para desplegar los cambios está restringida por reglas de acceso, el virtual firewall (AWS Security Groups) sólo permite el ingreso a través del puerto 22 protegido por una llave privada, los únicos puertos expuestos de salida son el 80 y 443.

La autentificación en el servidor para el cliente que solicite consumir la API se maneja a través de sesiones guardadas localmente por cookies. Todo el tráfico es forzado a ser enviado por el puerto 443 para ser encriptado incluyendo las conexiones en tiempo real.

Como resultado de una evaluación de pares dentro del marco de la asignatura de Proyecto Final, fue recibida una evaluación del prototipo. Se utilizó el estándar ISO 15504 y la escala de likert para evaluar el resultado final del prototipo y el alcance de distintos objetivos del proyecto. El resultado se puede apreciar en [fig. 8].



1. Resultado de evaluación en escala de likert

# Conclusiones

Dentro de los objetivos específicos con respecto a la revisión bibliográfica de la literatura esta se realizó en varias fases, primero se realizaron búsquedas basadas en los términos claves, luego se filtraron los resultados con el objetivo de obtener los más recientes (artículos e investigaciones de 2016 al presente 2020), posterior a esto se leyeron los abstract de cada artículo con el objetivo de obtener un primer vistazo acerca del contenido e inferir si potencialmente este podía contribuir al proyecto, por último se leyeron en profundidad los artículos seleccionados.

Para el objetivo de la creación del dataset de tweets colombianos en español, se desarrollaron los scripts necesarios para la obtención, preprocesamiento y procesamiento en AWS Comprehend, estos están escritos en python 3 y las instrucciones para su correcta ejecución se encuentran anexas en el repositorio privado del proyecto.

Respecto al objetivo de la implementación de la red neuronal que permita clasificar tweets nuevos, ésta fue implementada correctamente usando Pytorch, y entrenada con el dataset creado previamente, obteniendo como resultado predicciones con margen de error del 21.6%. [fig. 14]

Para la obtención de los tweets de las capitales de los departamentos de Colombia se requirió de la elaboración de una lista con información detallada de la localización de cada capital para los 32 departamentos de Colombia esto se logró usando el API de Google Maps, la cual provee las coordenadas, y también los límites de cada lugar, permitiendo ajustar los rangos de búsqueda. Esta lista se encuentra anexada en el repositorio privado en formato JSON.

Con respecto a la visualización de un mapa de calor para mostrar los resultados de la clasificación de sentimientos de las 32 capitales de los departamentos de Colombia, se implementó exitosamente en el client-side usando el componente GoogleMap para integrar Google Maps en la página de búsqueda y HeatmapLayer para añadir un layer de mapa de calor sobre el componente previamente mencionado. Además gracias a la sugerencia del tutor del curso Wilson Nieto, también se puede acceder a los resultados de los datos por medio de gráficas de pie usando el componente ChartJS.

Con los objetivos específicos cumplidos a cabalidad se concluye que el se ha cumplido el objetivo general de la presente investigación, el cual es el desarrollo de una aplicación que permita clasificar los sentimientos sobre productos, servicios o personas a través de tweets utilizando técnicas de machine learning y visualización de datos. Se puede acceder a esta através del siguiente enlace [http://sentime.com.co](http://sentime.com.co/)

Como objeto de una siguiente investigación se tienen las siguientes consideraciones en el marco de disminuir el margen de error de la red neuronal desarrollada:

* Para generar un dataset de tweets representativo de la jerga local se requiere de gran capacidad de cómputo, a medida de que este enriquezca el vocabulario del corpus mejorará el porcentaje de acierto de la red.
* Debe tenerse en cuenta el porcentaje error provisto por la clasificación del dataset utilizando Amazon Comprehend, se estima que es un 6.8% si está siendo utilizado el algoritmo BERT (que posee el mejor porcentaje de acierto en la actualidad 93.2% F1 para NLP).
* Al analizar la matriz de confusión [fig. 10] encontramos que las combinaciones entre positivos-neutral y negativos-neutral acumulan la mayor cantidad de falsos positivos, por lo que puede ser considerado remover esta categoría

Para el caso de el desarrollo de la plataforma web se tiene la siguiente consideración:

* El tamaño de los modelos en memoria sumado a el procesamiento en paralelo de las búsquedas por multihilos requieren de servidores escalables horizontalmente para contar con buen desempeño para alta concurrencia de usuarios.

Los datos resultantes de las búsquedas efectuadas en sentime pueden ser procesados en otras plataformas. Un ejemplo de esto puede ser un DSS que utilice los resultados de los sentimientos que tenga cada capital de los departamentos para asistir en la toma de decisión de ciertas acciones que pueda necesitar una empresa en un momento dado, en base a la reacción que tenga la gente acerca de cierto producto o servicio que ellos ofrezcan en el momento o tengan pensado ofrecer.

##### Referencias

1. Casero-Ripollés, A. (2020). Influencia de los medios de comunicación en la conversación política en Twitter. Revista ICONO14 Revista Científica de Comunicación y Tecnologías Emergentes, 18(1), 33–57. https://doi.org/10.7195/ri14.v18i1.1527
2. Puertas-Hidalgo, R., Carpio-Jimenez, L., & Suing, A. (2019). Political Communication and Social Networks. Twitter: Comparative Analysis of the Presidencies of the Republic of Latin America. 2019 14th Iberian Conference on Information Systems and Technologies (CISTI). doi:10.23919/cisti.2019.8760878
3. https://mintic.gov.co/portal/inicio/Sala-de-Prensa/Sabia-Ud-que/2713:Colombia-es-uno-de-los-paises-con-mas-usuarios-en-redes-sociales-en-la-region
4. Hernández, Iván (2019, Mayo 19) La renovación de las investigaciones de mercado, más allá del universo digital. Recuperado de https://www.rcnradio.com/tecnologia/la-renovacion-de-las-investigaciones-de-mercado-mas-alla-del-universo-digital
5. Villalobos, Carlos (2019, July 29) La importancia de las PYMES en Colombia. Recuperado de https://blog.hubspot.es/marketing/la-importancia-de-las-pymes-en-colombia.
6. Redacción Economía (2018, Marzo 16). El 62% de las pymes colombianas no tiene acceso a financiamiento. Recuperado de https://www.elespectador.com/economia/el-62-de-las-pymes-colombianas-no-tiene-acceso-financiamiento-articulo-744870.
7. Twintproject, twint, (2020), GitHub repository, https://github.com/twintproject/twint.
8. Peter Turney (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the Association for Computational Linguistics. pp. 417-424. arXiv:cs.LG/0212032.
9. Pang, B., Lee, L., Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. www.cs.cornell.edu.
10. Soni, A. K. (2017). Multi-lingual sentiment analysis of Twitter data by using classification algorithms. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). doi:10.1109/icecct.2017.8117884
11. Kaladevi, P., & Thyagarajah, K. (2019). Integrated CNN- and LSTM-DNN-based sentiment analysis over big social data for opinion mining. Behaviour & Information Technology, 1–9. doi:10.1080/0144929x.2019.1699960
12. Zhang, J., Li, Y., Tian, J., & Li, T. (2018). LSTM-CNN Hybrid Model for Text Classification. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 1675–1680. https://doi.org/10.1109/IAEAC.2018.8577620
13. Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment Analysis of Comment Texts Based on BiLSTM. IEEE Access, 7, 51522–51532. https://doi.org/10.1109/ACCESS.2019.2909919
14. Ceron-Guzman, J. A., & Leon-Guzman, E. (2016). A Sentiment Analysis System of Spanish Tweets and Its Application in Colombia 2014 Presidential Election. 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom). doi:10.1109/bdcloud-socialcom-sustaincom.2016.47
15. Chen, B., Huang, Q., Chen, Y., Cheng, L., & Chen, R. (2018). Deep Neural Networks for Multi-class Sentiment Classification. 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). doi:10.1109/hpcc/smartcity/dss.2018.00142
16. Paredes-Valverde, M. A., Colomo-Palacios, R., Salas-Zárate, M. del P., & Valencia-García, R. (2017). Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach. Scientific Programming, 2017, 1–6. doi:10.1155/2017/1329281
17. Suresh, H., & Gladston Raj S. (2016). An unsupervised fuzzy clustering method for twitter sentiment analysis. 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). doi:10.1109/csitss.2016.7779444
18. Parveen, H., & Pandey, S. (2016). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). doi:10.1109/icatcct.2016.7912034
19. Kusrini, & Mashuri, M. (2019). Sentiment Analysis In Twitter Using Lexicon Based and Polarity Multiplication. 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT). doi:10.1109/icaiit.2019.8834477
20. Huacon, C. F., & Pelegrin, L. (2017). SURV: A system for massive urban data visualization. 2017 IEEE MIT Undergraduate Research Technology Conference (URTC). doi:10.1109/urtc.2017.8284174
21. Diouf, R., Sarr, E. N., Sall, O., Birregah, B., Bousso, M., & Mbaye, S. N. (2019). Web Scraping: State-of-the-Art and Areas of Application. 2019 IEEE International Conference on Big Data (Big Data). doi:10.1109/bigdata47090.2019.9005594
22. Pytorch, pytorch, (2020), GitHub repository, https://github.com/pytorch/pytorch
23. Pallets, Flask, (2020), GitHub repository, https://github.com/pallets/flask/
24. SocketIO, Socket.IO, (2020), GitHub repository, https://github.com/socketio/socket.io
25. Google, Google Maps, (2020), Documentation Webpage, https://developers.google.com/maps/documentation/javascript/
26. Chartjs, chart.js, (2020), GitHub repository, https://github.com/chartjs/Chart.js
27. Antirez, redis, (2020), GitHub repository, https://github.com/antirez/redis
28. Postgres, postgres, (2020), GitHub repository, https://github.com/postgres/postgres/
29. Explosion, spacy, (2020), GitHub repository, https://github.com/explosion/spaCy
30. Dccuchile, spanish-word-embeddings, (2020), GitHub repository, https://github.com/dccuchile/spanish-word-embeddings
31. Facebook, react, (2020), GitHub repository, https://github.com/facebook/react
32. Nginx, nginx, (2020), GitHub repository, https://github.com/nginx/nginx
33. Certbot, certbot, (2020), GitHub repository, https://github.com/certbot/certbot