



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**  
**CARRERA DE INENIERÍA EN SISTEMAS**  
**COMPUTACIONALES**

**TEMA:**

Influencia de las expresiones idiomáticas propias de una jerga  
sobre algoritmos de Análisis de Sentimientos.

**AUTOR:**

Vásconez Yulán , Julio Oswaldo

**Trabajo de titulación previo a la obtención del grado de**  
**Ingeniero en Sistemas Computacionales**

**TUTOR:**

Ing. Molina Flores, Gustavo Andrés, Mgs.

Guayaquil, Ecuador

21 de Marzo de 2017



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**

**CARRERA DE INENIERÍA EN SISTEMAS COMPUTACIONALES**

**CERTIFICACIÓN**

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Vásquez Yulán, Julio Oswaldo**, como requerimiento para la obtención del Título de **Ingeniero en Sistemas Computacionales**.

**TUTOR**

f. \_\_\_\_\_

**Ing. Molina Flores, Gustavo Andrés, Mgs.**

**DIRECTORA DE LA CARRERA**

f. \_\_\_\_\_

**Ing. Guerrero Yépez, Beatriz del Pilar, Mgs.**

**Guayaquil, a los 21 días del mes de Marzo del año 2017**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**

**CARRERA DE INENIERÍA EN SISTEMAS COMPUTACIONALES**

**DECLARACIÓN DE RESPONSABILIDAD**

**Yo, Vásconez Yulán, Julio Oswaldo**

**DECLARO QUE:**

El Trabajo de Titulación, **Influencia de las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimientos** previo a la obtención del Título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 21 días del mes de Marzo del año 2017

**EL AUTOR (A)**

f. \_\_\_\_\_

**Vásconez Yulán, Julio Oswaldo**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**

**CARRERA DE INENIERÍA EN SISTEMAS COMPUTACIONALES**

**AUTORIZACIÓN**

**Yo, Vásconez Yulán, Julio Oswaldo**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación, **Influencia de las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimientos**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 21 días del mes de Marzo del año 2017

**EL AUTOR:**

f. \_\_\_\_\_

**Vásconez Yulán, Julio Oswaldo**

## Urkund Analysis Result

**Analysed Document:** Tesis - Julio Vásconez.docx (D26008238)  
**Submitted:** 2017-02-24 22:07:00  
**Submitted By:** jvasconez28@gmail.com  
**Significance:** 1 %

### Sources included in the report:

[https://doi.org/10.5209/rev\\_ESMP.2012.v18.40980](https://doi.org/10.5209/rev_ESMP.2012.v18.40980)  
<http://www.forbes.com.mx/la-importancia-de-los-sentimientos-en-redes-sociales/>

### Instances where selected sources appear:

2

## **AGRADECIMIENTO**

Quiero agradecer a Dios por su guía y por cada una de las personas que puso en mi camino antes y a lo largo de este logro. A mi abuelita Lidia, mi mayor ejemplo de firmeza, decisión e ímpetu, que aunque no llegó a contemplar este logro se mantuvo presente en cada paso. A mis padres, Julio y Katyhusca, por el amor que me han brindado cada día de mi vida y el apoyo incondicional mostrado para alcanzar esta meta; por ser ambos ejemplo de esfuerzo, humildad. A mis hermanos: Alba, Adriana y Daniel, por alentarme en este camino con sus palabras, con su ejemplo y compañía. A mi enamorada, Andrea, por compartir mis anhelos y motivarme a cumplir mis metas. A mi tutor Ing. Gustavo Molina, por aportar con sus conocimientos y experiencia a la materialización de este proyecto. A todas y cada una de las personas que se aportaron a la construcción de mi carrera: amigos, compañeros, profesores.

Julio Oswaldo Vásconez Yulán

## **DEDICATORIA**

Dedico este trabajo a mi abuelita Lidia, que en el cielo debe alegrarse por saberme alcanzando esta meta, a mis padres porque este logro no es sólo mío, sino también de ellos ya que sólo ha sido posible por su esfuerzo y amor, a mis hermanos y mi enamorada por ser mis compañeros incondicionales en el camino a cumplir mis metas.

Julio Oswaldo Vásconez Yulán



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**

**CARRERA DE INENIERÍA EN SISTEMAS COMPUTACIONALES**

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

Ing. Molina Flores, Gustavo Andrés, Mgs.

**TUTOR**

f. \_\_\_\_\_

Ing. Guerrero Yépez, Beatriz del Pilar, Mgs.

**DIRECTORA DE CARRERA**

f. \_\_\_\_\_

Ing. Celleri Mujica, Colón Mario, Mgs

**COORDINADOR DEL ÁREA**

f. \_\_\_\_\_

Ing. Camacho Coronel, Ana Isabel, Mgs

**OPONENTE**





UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA**

**CARRERA DE INENIERÍA EN SISTEMAS COMPUTACIONALES**

**CALIFICACIÓN**

f. \_\_\_\_\_

Ing. Molina Flores, Gustavo Andrés, Mgs.

**TUTOR**

f. \_\_\_\_\_

Ing. Guerrero Yépez, Beatriz del Pilar, Mgs.

**DIRECTORA DE CARRERA**

f. \_\_\_\_\_

Ing. Celleri Mujica, Colón Mario, Mgs

**COORDINADOR DEL ÁREA**

f. \_\_\_\_\_

Ing. Camacho Coronel, Ana Isabel, Mgs

**OPONENTE**

# ÍNDICE

CAPÍTULO 1: Introducción .....	17
1.1 Problema .....	18
1.2 Preguntas de investigación.....	18
1.3 Objetivos .....	18
1.3.1 Objetivo General.....	18
1.3.2 Objetivos Específicos .....	18
1.4 Justificación .....	19
1.5 Alcance .....	19
CAPÍTULO 2: FUNDAMENTACIÓN CONCEPTUAL .....	20
2.1 Antecedentes .....	20
2.2 Lenguaje, Subjetividad y Cultura .....	21
2.3 Análisis de Sentimiento .....	21
2.3.1 Análisis de Sentimiento Basado en Léxicos.....	22
2.3.2 Machine Learning .....	22
2.3.3 Procesamiento de Lenguaje Natural .....	22
2.3.4 Relación Análisis de Sentimiento – Machine Learning .....	22
2.4 Componentes y Herramientas .....	23
2.4.1 Lenguaje de Programación: Python.....	23
2.4.2 Natural Language Toolkit.....	23
2.4.3 Clasificador Naive Bayes .....	24
2.4.4 Léxicos de Entrenamiento .....	25
CAPÍTULO 3: METODOLOGÍA DE INVESTIGACIÓN.....	26
3.1 Enfoque Metodológico.....	26
3.2 Población y Muestra .....	27
3.3 Técnicas e Instrumentos de Investigación .....	28
3.3.1 Observación directa .....	28
3.3.2 Encuestas .....	29
3.4 Análisis de Encuestas.....	29
3.5 Análisis de Observación Directa.....	33
3.6 Consolidado de Información.....	36
CAPÍTULO 4: PROPUESTA.....	37

4.1 Viabilidad Técnica .....	37
4.2 Descripción del Software .....	37
4.2.1 Python 3 .....	37
4.2.2 NLTK.....	38
4.2.3 Serializador de Objetos.....	38
4.2.4 Requests.....	39
4.3 Detalles de la Propuesta .....	39
4.3.1 Estructura de carpetas .....	39
4.3.2 Funcionalidades de librerías creadas .....	41
4.3.3 Construcción del Prototipo .....	44
4.3.4 Diagrama de flujo de Análisis de Sentimiento .....	47
4.4 Análisis de Resultados .....	47
CAPÍTULO 5: CONCLUSIONES .....	53
Bibliografía	54

## ÍNDICE DE TABLAS

Tabla 1. Definiciones en la Quintupla de la opinión.....	20
Tabla 2. Ejemplo de Quintupla de Opinión .....	20
Tabla 3: Población del caso de estudio .....	27
Tabla 4. Resultados observación directa para expresiones neutras y positivas .....	34
Tabla 5. Resultados observación directa para expresiones negativas.....	35
Tabla 6. Descripción de los módulos de clasificadores. ....	45
Tabla 7. Cantidad de expresiones reconocidas. ....	50
Tabla 8. Aceptación según clasificadores. ....	51
Tabla 9. Precisión de algoritmos básicos, comparados con los resultados de los algoritmos mejorados. ....	52

## ÍNDICE DE FIGURAS

Figura 1. Ranking de Lenguajes de Programación según la IEEE. <a href="http://programacion.net">http://programacion.net</a> .....	23
Figura 2. Teorema de Bayes. (Mendenhall, Beaver, & Beaver, 2009, p. 160) .....	24
Figura 3. Bayes Aplicado a Clasificación de Texto (“Naive Bayes text classification,” s.f.)	24
Figura 4. Ejemplo de construcción de árbol de probabilidades usando el algoritmo Naive Bayes. Elaborado por Autor. ....	25
Figura 5. Formato de recopilación de datos de Observación Directa .....	29
Figura 6. Expresiones Altamente Positivas más utilizadas. <b>Elaborado por</b> Autor.	29
Figura 7. Expresiones Positivas más utilizadas. Elaborado por Autor. ....	30
Figura 8. Expresiones Neutras más utilizadas Elaborado por Autor. ....	31
Figura 9. Expresiones Negativas más utilizadas Elaborado por Autor. ....	32
Figura 10. Expresiones Negativas más utilizadas Elaborado por Autor. ....	33
Figura 11. Estructura archivo de léxicos. Elaborado por Autor. ....	36
Figura 12. Estructura de Carpetas. Elaborado por Autor. ....	39
Figura 13. Detalles archivo de inicialización e importación python. Elaborado por Autor.	40
Figura 14. Pseudocódigo obtención y paginación de Tweets. Elaborado por Autor.	41
Figura 15. Captura de código del módulo de obtención de Tweets. Elaborado por Autor.	41
Figura 16. Pseudocódigo del módulo de entrenamiento de un Clasificador Bayesiano. 42	
Figura 17. Captura del código de un clasificador Binario. Elaborado por Autor. ...	42
Figura 18. Captura del módulo de Procesamiento de Lenguaje Natural para el idioma Español. Elaborado por Autor. ....	43
Figura 19. Estructura de carpeta de código fuente. Elaborado por Autor. ....	44
Figura 20. Captura del código de la función de clasificación de texto. Elaborado por Autor.	46
Figura 21. Captura de resultados de ejecución de prototipo. Elaborado por Autor.	46
Figura 22: Diagrama de Procesos .....	47

Figura 23. Resultados en consola de la ejecución del prototipo. Elaborado por Autor.	
48	
Figura 24. Visualizaciones de tendencia de Sentimiento generadas por el prototipo. Elaborado por Autor.....	48
Figura 25. Captura de resultados de ejecución de clasificador multiclase. Elaborado por Autor. ....	49
Figura 26. Captura de líneas de tendencia, resultados de clasificador multiclase. Elaborado por Autor.....	49

## RESUMEN

Considerando lo importante que puede llegar a ser para una persona, institución o producto, en la actualidad; conocer el nivel de aceptación o el sentimiento que genera en las en la sociedad digital, surge el *Análisis de Sentimiento* como respuesta, con la finalidad de procesar una gran cantidad de opiniones de manera automática, haciendo uso de técnicas de clasificación.

Dentro de la problemática a la que se enfrenta la disciplina mencionada se tiene el contexto que se desprende de las diferente granularidades geográficas como diferencia de continente, región, país, ciudades, e incluso, sectores dentro de las ciudades; a nivel lingüístico. No solamente el idioma afecta gravemente a los resultados de este Análisis, sino también *las expresiones idiomáticas propias de la jerga* de cada zona geográfica o conjunto social.

El desarrollo de un prototipo de Análisis de Sentimiento basado en léxicos permite enfrentar la problemática mencionada al añadir expresiones idiomáticas propias de la jerga ecuatoriana a las consideraciones del algoritmo de clasificación, mismas que fueron recopiladas a través de una investigación cualitativa dentro de una muestra de usuarios digitales de Ecuador. Luego de ejecutar y comparar los resultados de los algoritmos mejorados ha sido posible describir la incidencia de la adición del léxico ecuatoriano dentro de la precisión de los algoritmos de Análisis de Sentimiento.

**Palabras Clave:** SENTIMIENTO, OPINIÓN, ANÁLISIS, SUBJETIVIDAD, APRENDIZAJE-AUTOMÁTICO, LÉXICO.

## ABSTRACT

Given the importance of the public opinion and acceptance levels, for people, institutions or products, nowadays. The Sentiment Analysis raises as the answer, with the objective of classify massive opinions automatically, using Machine Learning classification algorithms.

This discipline (Sentiment Analysis) is facing a context problem when the geographical differences appears. This could affect directly the classification results. The problem is that the lexicon changes not just by the languages, it could changes by the idiomatic expressions of every continent, region, country, city or zone.

The development of a lexicon based Sentiment Analysis prototype gives as the tools to face and beat this context problem in Ecuador, constructing a sentiment dictionary from a qualitative research on digital Ecuadorian users. After we execute and analyze the results we can describe the incidence of adding Ecuadorian idiomatic expressions to the classification algorithms results.

**Keywords:** SENTIMENT, OPINION, ANALYSIS, SUBJECTIVITY, MACHINE-LEARNING, LEXICON.



## CAPÍTULO 1: Introducción

Las redes sociales se han convertido en un elemento de interacción entre personas de distintas partes del mundo, características, razas, profesiones, etc., y ello hace reconocer el gran valor que puede tener en la actualidad para una institución o personaje público, conocer el impacto conseguido en las redes sociales de opinión como Twitter, y, tal como indica Rubén Vázquez de Forbes México: “Medir la manera en que los usuarios de las redes sociales se expresan sobre una cuenta o un tema en específico es **fundamental** para no errar nuestros objetivos.” (2014, párr. 1). La sociedad se enfrenta a una creciente demanda de información visible y que sea posible interpretar con el objetivo de evaluar la imagen proyectada por la institución o personaje público en medios sociales, con la finalidad de apoyar la toma de decisiones, medidas y estrategias que les permitan dirigir su impacto de manera positiva. Casos aplicables de personas o instituciones que requieren conocer el impacto social que están provocando son: candidatos presidenciales o dignidades públicas en general; productos que tienen tiempo en el mercado; empresas en general, instituciones públicas; entre muchos otros casos.

Se convierte en un reto brindar a los demandantes la capacidad de obtener datos visibles y entendibles (Visualización de Datos), fruto del análisis de las opiniones extraídas (Minería de opiniones) de las redes dispuestas anteriormente. “La minería de opiniones (OM) o análisis de sentimientos está cobrando cada vez mayor importancia debido fundamentalmente a la gran cantidad de comentarios que se escriben en Internet por parte de millones de usuarios de todo el mundo.”(Martínez Cámara, Martín Valdivia, Perea Ortega, & Ureña López, 2011, p. 1). Un reto probablemente inalcanzable de manera manual. Incluso de gran complejidad si fuera automatizado mediante técnicas de programación imperativa. Razones por las cuales es necesario buscar soluciones en técnicas con mayor alcance.

Considerando, el avanzado desarrollo del aprendizaje automático (Machine Learning), y de manera específica del área de Procesamiento de Lenguaje Natural (Natural Language Processing) para el Análisis de Sentimiento (Sentiment Analysis). Es posible encontrar soluciones aplicables a las necesidades mencionadas respecto a un tema, institución o personaje. “Aunque existen ya varios trabajos relacionados con la temática, la mayoría de ellos únicamente usan textos en inglés.” (Martínez Cámara et al., 2011) Las herramientas existentes poseen una limitación, la mayoría han sido implementadas con base en la lengua inglesa y las pocas que han sido creadas para el idioma español no reconocen expresiones idiomáticas propias de la

jerga de un país como Ecuador, por lo que ofrecen Análisis de Tendencia de términos, mas no consiguiendo clasificar el sentimiento generado en los usuarios de manera que pueda visualizarse o tabularse.

## **1.1 Problema**

En la actualidad no existen aplicaciones que permitan realizar *Análisis de Sentimiento* que sean capaces de clasificar expresiones idiomáticas propias de la jerga ecuatoriana.

## **1.2 Preguntas de investigación**

1. ¿Cuáles son las expresiones idiomáticas propias de la jerga ecuatoriana?
2. ¿Cómo afectará la inclusión de reconocimiento de expresiones idiomáticas propias de la jerga ecuatoriana a un algoritmo de Análisis de Sentimiento existente?

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Implementar un prototipo de aplicación de código abierto para *Análisis de Sentimiento*, incluyendo soporte a expresiones idiomáticas propias de la jerga ecuatoriana para describir la influencia de las mismas sobre los resultados del algoritmo.

### **1.3.2 Objetivos Específicos**

- Identificar expresiones idiomáticas propias de la jerga ecuatoriana, haciendo uso de observación directa y encuestas digitales que puedan ser clasificadas en una escala según su intensidad;
- Diseñar algoritmo de minería de opiniones basado en algoritmos de Procesamiento de Lenguaje Natural existente, añadiendo expresiones idiomáticas de la jerga ecuatoriana;
- Describir la incidencia de la inclusión de expresiones idiomáticas en el algoritmo de Análisis de Sentimientos para comprobar la medida en que esto optimiza al algoritmo; y,
- Publicar los resultados y código del prototipo desarrollado a través GitHub para contribuir en el desarrollo del Análisis de Sentimiento e impulsar el desarrollo de código abierto.

## **1.4 Justificación**

Teniendo en cuenta la demanda global de Análisis de Sentimiento en gran variedad de casos, desde evaluación de situación electoral de actores políticos hasta dentro de estudios de mercado para verificar la aceptación de un producto, o el posicionamiento de una marca en las redes mencionadas con anterioridad. Es preciso brindar una solución aplicable a Ecuador; permitiendo, de esta manera, que personajes e instituciones tengan la posibilidad de realizar análisis de mayor exactitud respecto al sentir del país.

Por lo que se concluye en la existencia de una necesidad latente que justifica el desarrollo de este proyecto, teniendo como objeto la implementación de un prototipo que brinde una solución de código abierto a los demandantes de la misma.

## **1.5 Alcance**

Este proyecto de investigación será desarrollado en un tiempo de 16 semanas, en las que se pretende recopilar y analizar datos de los modismos de la lengua ecuatoriana mismos que serán obtenidos mediante observación directa y encuestas digitales, metodologías explicadas y sustentadas más adelante. Lo mencionado, se realizará con el fin de mejorar un algoritmo de Análisis de Sentimiento que será implementado en un prototipo de aplicación con el fin de comparar los resultados con el mismo algoritmo sin modificaciones.

Al concluir se publicará el código fuente del prototipo desarrollado en la plataforma de GitHub en la cuenta del autor (jvas28), bajo Licencia Pública General de GNU con el objetivo de contribuir al desarrollo de esta rama del Procesamiento del Lenguaje Natural en el país.

## CAPÍTULO 2: FUNDAMENTACIÓN CONCEPTUAL

### 2.1 Antecedentes

La información textual de todo el mundo digital puede ser clasificada de manera general en hechos y opiniones. Los hechos son expresiones objetivas sobre entidades, eventos y sus diversas propiedades. Las opiniones son normalmente expresiones subjetivas que describen el sentimiento de las personas, pudiendo ser positivos o negativos (Liu, 2010)

Liu en su libro “Sentiment Analysis and Opinion Mining” profundiza en la definición y estructura de la opinión la cual divide en cuatro partes, las dos principales son el objetivo (target), y el sentimiento (sentiment), finalmente agrega tres componentes para así definirla como una quintupla, el tiempo (time), la titular (holder) y la(s) característica(s) (attribute) del objetivo. (Gandecha, Gondane, & Shelke, s. f.)

**Tabla 1.** Definiciones en la Quintupla de la opinión

<b>Componente</b>	<b>Definición</b>
Objetivo	Entidad, o persona a la que se refiere la oración.
Característica	Característica o Atributo del Objetivo que genera el sentimiento.
Sentimiento	Es la descripción subjetiva de la experiencia causada por la característica del objetivo.
Tiempo	Es el espacio cronológico en que la opinión fue emitida.
Titular	Es la persona que emite la opinión

Elaborado por: Autor

**Tabla 2.** Ejemplo de Quintupla de Opinión

<b>Ejemplo</b>	
Julio Vásquez: La empresa XYZ brinda un excelente servicio de reparación de computadoras. 22/12/16	
<b>Componente</b>	<b>Correspondencia</b>
Objetivo	Empresa XYZ
Característica	Servicio de Reparación de Computadoras
Sentimiento	Excelente
Tiempo	22/12/2016
Titular	Julio Vásquez

Elaborado por: Autor

En las tablas 1 y 2 anteriores se puede apreciar de manera más clara la composición de una opinión, de la cual no siempre se tiene la información total, pero para el estudio, bastará poseer el **objetivo y el sentimiento**.

## **2.2 Lenguaje, Subjetividad y Cultura**

El lenguaje es un aspecto único y definitorio, mismo que le permitió al hombre diferenciarse de los demás seres de la tierra, haciendo uso este se convirtió en un productor y medio de transmisión de cultura. El lenguaje es organizado y estructurado por lo que permite al interlocutor apropiarse del mismo, hacer suyo lo que dice, es aquí donde la subjetividad aparece, cuando cualquiera de los interlocutores es capaz de apropiarse del lenguaje sólo agregando la palabra yo. La aparición de esta “Subjetividad” en el lenguaje es motivo de creación de la categoría de persona en el mismo lenguaje e incluso fuera del mismo (Liu, 2010)

## **2.3 Análisis de Sentimiento**

El Análisis de Sentimiento es un nuevo tipo de análisis de texto cuyo objetivo final es determinar el nivel de subjetividad que posee una opinión o comentario de usuarios que realizan comentarios sobre un tema o producto. Con la creciente popularidad de sitios como Amazon en lo que las personas pueden publicar opiniones y calificar productos, el internet se llena de comentarios, opiniones y calificaciones (Collomb, Costea, Joyeux, Hasan, & Brunie, s. f.).

También Collomb, explica los campos de aplicación de este Análisis considerándolo de total utilidad al brindar información a los clientes que les ayuda a tomar decisiones respecto a los productos según la reputación que tienen respecto a la opinión pública. El mismo puede revelar que piensan las personas en general de un producto. Desde la otra cara de la moneda es útil para los propietarios de productos para conocer la reputación que están generando en la red. Finalmente el mismo fue propuesto para analizar largos textos y eliminar las partes más subjetivas de los mismos y presentar publicidad según las opiniones que el usuario ha expuesto (s.f).

Se considera esta disciplina un problema de clasificación en muchos casos de 2 clases: Sentimiento Positivo, Sentimiento Negativo. Al ser un problema tradicional de clasificación de texto es posible aplicar cualquier método de aprendizaje supervisado puede ser aplicado. (Gandecha et al., s. f.)

### **2.3.1 Análisis de Sentimiento Basado en Léxicos**

El enfoque basado en léxicos envuelve el cálculo de la orientación semántica basado en un documento que contiene la orientación semántica de las palabras o frases. Lo que lo convierte en un problema de clasificación (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011). Es necesario poseer un documento para la ejecución de Análisis de Sentimiento según este enfoque, mismo que permita entrenar un algoritmo para realizar la clasificación.

### **2.3.2 Machine Learning**

El aprendizaje, como la inteligencia, abarca una amplia gama de procesos que no son fáciles de definir precisamente, el diccionario lo define como *obtener conocimiento o entendimiento o habilidades a través de estudio o experiencia*. El Aprendizaje de Automático (Machine Learning) suele referirse a los cambios en los sistemas que realizan tareas de Inteligencia Artificial, tareas como reconocimiento, diagnóstico, planeación, control de robots, predicción, etc.

#### **2.3.2.1 Aprendizaje Supervisado**

Ciertamente el aprendizaje es un proceso bastante complejo por lo que ha surgido la necesidad, en el Aprendizaje Automático, dividirlo según la tarea de aprendizaje que se realice (Shalev-Shwartz & Ben-David, 2014). El aprendizaje supervisado a breves rasgos parte de resultados conocidos que permitan entrenar al algoritmo para responder ante características similares.

### **2.3.3 Procesamiento de Lenguaje Natural**

El Procesamiento de Lenguaje Natural comprende un conjunto de técnicas para realizar análisis y representar de manera natural los textos en uno a más niveles del análisis lingüístico con el propósito de alcanzar la comprensión del lenguaje humano para el desarrollo de tareas. (Liddy, 2001)

### **2.3.4 Relación Análisis de Sentimiento – Machine Learning**

El Análisis de Sentimiento y el Aprendizaje Automático, entablan una relación, inicialmente, a través de la necesidad de clasificar texto. Una de las técnicas más comunes para realizar clasificación es hacer uso de métodos de aprendizaje supervisado como clasificadores. Los más utilizados a nivel de Análisis de Sentimiento son los **Naive Bayes** y las Maquinas de Soporte Vectorial (SVM, por sus siglas en inglés).

## 2.4 Componentes y Herramientas

### 2.4.1 Lenguaje de Programación: Python

Este lenguaje fue creado por Guido Van Rossum a finales de los 80, diferenciándose de los lenguajes populares como C, C++ y Java, por la sencillez de su sintaxis, pero a la vez por su funcionalidad (Donzeau-Gouge, Huet, Kahn, & Lang, 1980, p. 2).

Es de saber público que grandes empresas de tecnología como Google o Yahoo o instituciones de alta importancia como la NASA o el CERN han optado por añadir Python entre los lenguajes para desarrollo interno ya que su **hermosa sintaxis y paradigma funcional** permiten el desarrollo de potentes aplicaciones en mucho menos líneas de código que los demás lenguajes.

Por las razones mencionadas, Python toma gran popularidad entre los usuarios en la actualidad ubicándose en el top 5 de lenguajes de programación del 2016 según la IEEE.



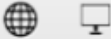





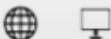
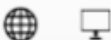
Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Figura 1. Ranking de Lenguajes de Programación según la IEEE. <http://programacion.net>

### 2.4.2 Natural Language Toolkit

El Kit de Herramientas de Procesamiento de Lenguaje Natural de Python es una librería de dominio específico que permite a los desarrolladores realizar tareas de procesamiento de lenguaje natural con facilidad. La misma brinda facilidades al momento de realizar 2 procesos

fundamentales para el Análisis de Sentimiento la disgregación de las palabras (Tokenización) y la extracción de las raíces de las palabras (Stemming).

### 2.4.3 Clasificador Naive Bayes

También conocido como clasificador ingenuo de Bayes, es un algoritmo de Aprendizaje Supervisado que debe ser entrenado con datos conocidos, asignando probabilidades de a los datos conocidos con relación al teorema de Bayes.

#### 2.4.3.1 Teorema de Bayes

El teorema de Bayes es una simple fórmula matemática utilizada para realizar cálculos de probabilidad condicional (Joyce, 2016). Se utiliza para revisar probabilidades previamente calculadas cuando se poseen nuevos datos.

$$P(S_i|A) = \frac{P(S_i)P(A|S_i)}{\sum_{j=1}^k P(S_j)P(A|S_j)}$$

para  $i = 1, 2, \dots, k$ .

Figura 2. Teorema de Bayes. (Mendenhall, Beaver, & Beaver, 2009, p. 160)

La fórmula describe la probabilidad de que un Suceso  $i$ , sea el predecesor de  $A$  en el árbol de probabilidades, o la causa. En otras palabras podríamos decir que el teorema de Bayes sirve para definir la existencia de una relación Causa-Efecto conociendo datos históricos que permitan asignar probabilidades relacionando la probabilidad de Efecto dado Causa y Causa dado efecto. (Parzen, 1971)

#### 2.4.3.2 Aplicación en Clasificación de Sentimientos

El clasificador Naive Bayes construye un árbol de probabilidades a partir de los datos de entrenamiento conocidos como **documento (d)**, para luego calcular la probabilidad de pertenencia a cada **clase (c)**.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Figura 3. Bayes Aplicado a Clasificación de Texto («Naive Bayes text classification», s. f.)



La fórmula de la *Figura 3* responde al teorema de Bayes aplicado sobre la clasificación de texto, de la siguiente manera. Permitiendo obtener la probabilidad de que el documento  $d$  pertenezca a una clase  $c$ . A continuación se ilustra un ejemplo de datos de entrenamiento y la construcción de un árbol de probabilidades.

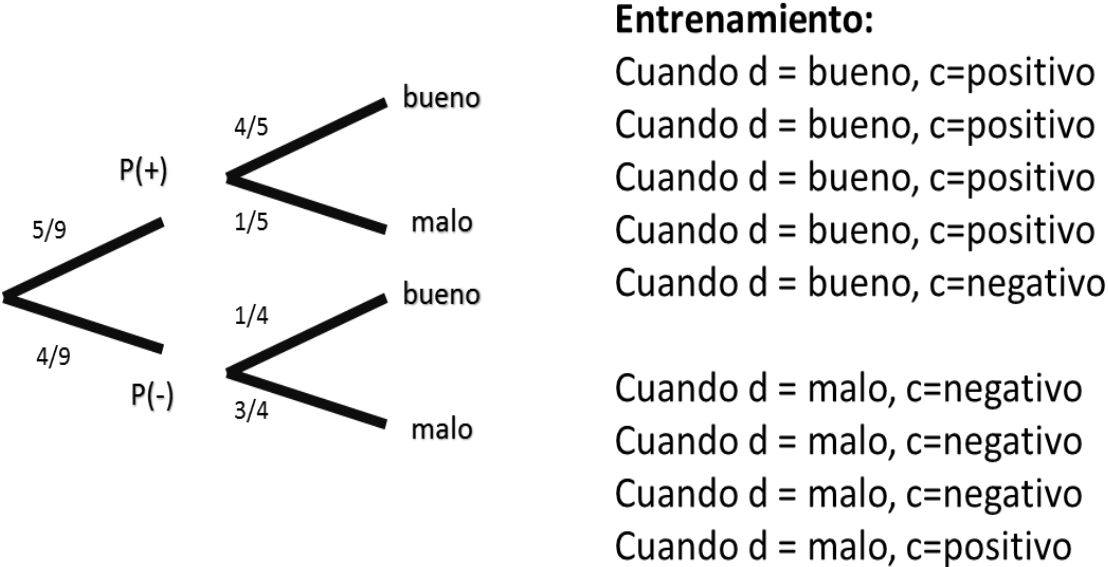


Figura 4. Ejemplo de construcción de árbol de probabilidades usando el algoritmo Naive Bayes.

Elaborado por Autor.

**2.4.4 Léxicos de Entrenamiento**

En la web existen disponibles recursos de entrenamiento, en su gran mayoría para el idioma inglés, pero algunas Universidades y Asociaciones han construido léxicos que pueden ser utilizados para estudios bajo licencias no comerciales. Estos diccionarios de datos brindan un listado de palabras con un valor de sentimiento, en general positivo y negativo, pero también existen en escalas.

## CAPÍTULO 3: METODOLOGÍA DE INVESTIGACIÓN

### 3.1 Enfoque Metodológico

Considerando que la investigación se constituye de una secuencia de procesos, sistemáticos, críticos y empíricos (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2014, p. 4) razón por la que una metodología debe ser definida para la correcta realización de la misma. Se ha de optar por el enfoque que se ajuste a los objetivos del proyecto como se describe a continuación:

El enfoque cualitativo también se guía por áreas o temas significativos de investigación. Sin embargo, en lugar de que la claridad sobre las preguntas de investigación e hipótesis preceda a la recolección y el análisis de los datos (como en la mayoría de los estudios cuantitativos), los estudios cualitativos pueden desarrollar preguntas e hipótesis antes, durante o después de la recolección y el análisis de los datos. Con frecuencia, estas actividades sirven, primero, para descubrir cuáles son las preguntas de investigación más importantes; y después, para perfeccionarlas y responderlas (Hernández Sampieri et al., 2014, p. 7).

Esta investigación tuvo un **enfoque cualitativo**, puesto que su finalidad, dentro del proyecto, será recopilar y clasificar expresiones idiomáticas utilizadas en la jerga ecuatoriana que califiquen un sentimiento según su intensidad para **describir** el modo en el que afecta la adición de soporte a expresiones idiomáticas propias de la jerga ecuatoriana a un algoritmo existente de Análisis de Sentimiento. Motivo por el que se optó por métodos de recolección de información como **observación directa** y **encuestas digitales** respectivamente, resultados que fueron cruzados para obtener un consolidado final de expresiones y sus respectivos valores según su intensidad. Datos que luego se convierten en entradas para el algoritmo de Análisis de Sentimientos.

La observación directa se realizó sobre comportamientos y mensajes de usuarios de twitter en general, refiriéndose a tendencias actuales (Trendig Topics), y evaluando servicios de establecimientos en Twitter, con el fin de recopilar los modismos de los ecuatorianos al referirse a tópicos específicos.

Va incrementándose la frecuencia con la que se usa la web como herramienta de soporte para la investigación y el análisis de datos (P. de Marchis, 2012, p. 1). Se encuentra ventajas al utilizar métodos de investigación por internet, entre las que se puede destacar: la amplificación de los alcances de la investigación al eliminar las fronteras geográficas; facilidad

de acceder a especialistas en lugares remotos; una relación costo – beneficio beneficiosa; también la eficiencia de los mismos al tomar mucho menos tiempo y automatizar la tabulación (Hewson, Vogel, & Laurent, 2016, p. 38). Finalmente es posible encontrar un beneficio propio de los sistemas que es la integridad que brindan las herramientas web al eliminar del proceso la transcripción de encuestas que está expuesto a errores. Por las afirmaciones anteriores y considerando que la población objetiva de este estudio son usuarios digitales (Usuarios de Twitter), se ha concluido en el uso de encuestas digitales como segundo instrumento de recolección de información.

Fruto de la información recolectada se obtendrá las expresiones idiomáticas de la jerga ecuatoriana y sus niveles de intensidad que serán añadidos al algoritmo con el fin de comparar los resultados y describir la incidencia de este proceso.

### 3.2 Población y Muestra

A las 23h00 del 6 de noviembre del año 2016, Ecuador tenía una población de 16.618.606 personas (INEC, 2016) dato que es el punto de partida inicial para la definición de la muestra del proyecto, como se presenta a en la tabla 3.

**Tabla 3:** Población del caso de estudio

Descripción	Porcentaje	Total
Población Total	100,00	16.618.606
Posee cuenta de redes sociales	41,50	6.896.721
Posee cuenta de Twitter	8,50	586.221

Fuente: (INEC, s. f.). Elaborado por: Autor

Partiendo de la población total del país, mencionada en el párrafo anterior, se obtiene que el 41,5% de los ecuatorianos poseen cuentas de redes sociales y de este número 8,5% posee cuenta de twitter dejando como población objetiva del caso de estudio a 586.221 personas que cumplen con ser usuarios de la red Social Twitter. Con estos datos se aplica la siguiente ecuación propuesta por Bolaños Rodríguez (2012).

$$n = \frac{N \cdot Z^2 \cdot p \cdot (1-p)}{(N-1) \cdot e^2 + Z^2 \cdot p \cdot (1-p)}$$

**Fuente:** (Dr. Ernesto Bolaños Rodríguez, 2012)

**Elaborado por:** Autor

Donde **n** es el tamaño de la muestra (la cantidad de encuestas a realizarse para el estudio); **N** es la población objeto del estudio (Usuarios de twitter en Ecuador); **Z** es la desviación que se acepta para lograr el nivel de confianza deseado que para este estudio se estableció en 95%, por lo que es de 2.575; **e** es el margen de error admitido en este estudio su valor fue de 5% ; **p** es la proporción que se esperaba encontrar, cuyo porcentaje es de 50% cuando no se tiene información de la proporción esperada.

El resultado obtenido al realizar el cálculo de la muestra haciendo uso de la ecuación y valores mencionados anteriormente fue de **384 encuestas** a realizarse, dentro de la población que objeto del estudio: los usuarios de twitter de Ecuador.

### **3.3 Técnicas e Instrumentos de Investigación**

Como anteriormente fue mencionado, las técnicas para recolección de información fueron la observación y la encuesta, el objetivo de ambas fue obtener una lista de términos que podrá ser usada como objeto de análisis para el algoritmo de análisis de sentimiento al evaluar las opiniones de los usuarios. Para la observación directa, que para el caso de este estudio fue ~~será~~ abierta, se diseñó una guía básica de aspectos importantes; el formato de encuesta digital, fue diseñado para obtener información que permita agilizar el proceso de recolección de expresiones idiomáticas usadas en Ecuador.

#### **3.3.1 Observación directa**

La observación directa, realizada de manera abierta, no sistemática, en la red social de twitter con la finalidad de obtener listas de expresiones dentro de la escala de sentimientos del 1 al 5, tomando como modelo la escala de Likert.

Pésimo	Malo	Neutro	Bueno	Excelente

Figura 5. Formato de recopilación de datos de Observación Directa

### 3.3.2 Encuestas

Obtenido el tamaño de la muestra en el apartado anterior, la encuesta a realizarse tendrá el mismo objetivo de la observación directa, en el **anexo 1** se puede observar una abstracción del diseño de la misma que fue digitalizada y enviada mediante internet a los usuarios del público objetivo que en su totalidad está compuesto por usuarios digitales. La estructura responde a la necesidad de obtener un léxico compuesto de expresiones y clasificadas según el valor de sentimiento.

### 3.4 Análisis de Encuestas

Los resultados de las encuestas realizadas, responden a la estructura para la construcción de un léxico, correspondiente a la jerga ecuatoriana, mismas que serán datos de entrenamiento para el algoritmo mejorado de Análisis de Sentimiento.

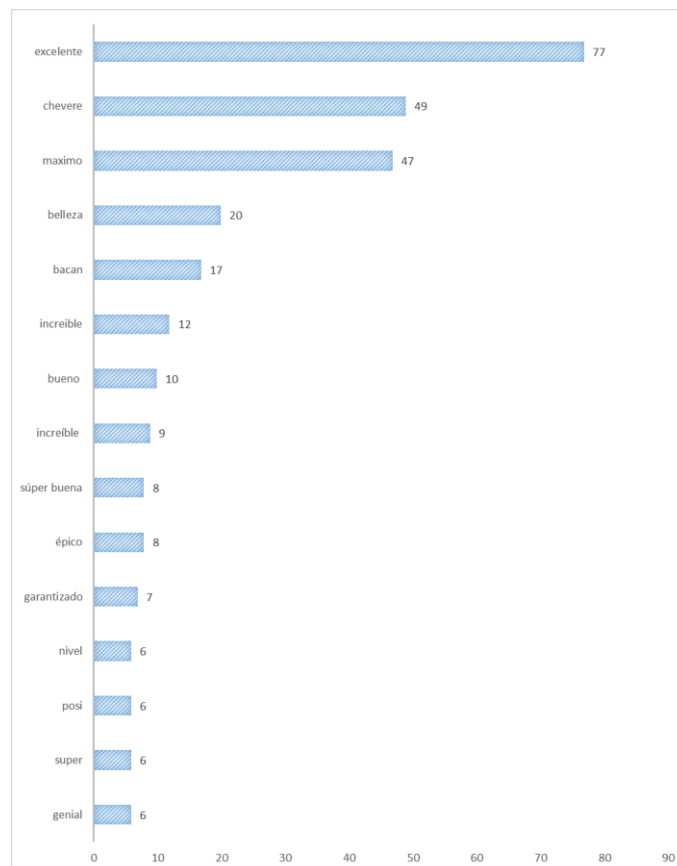
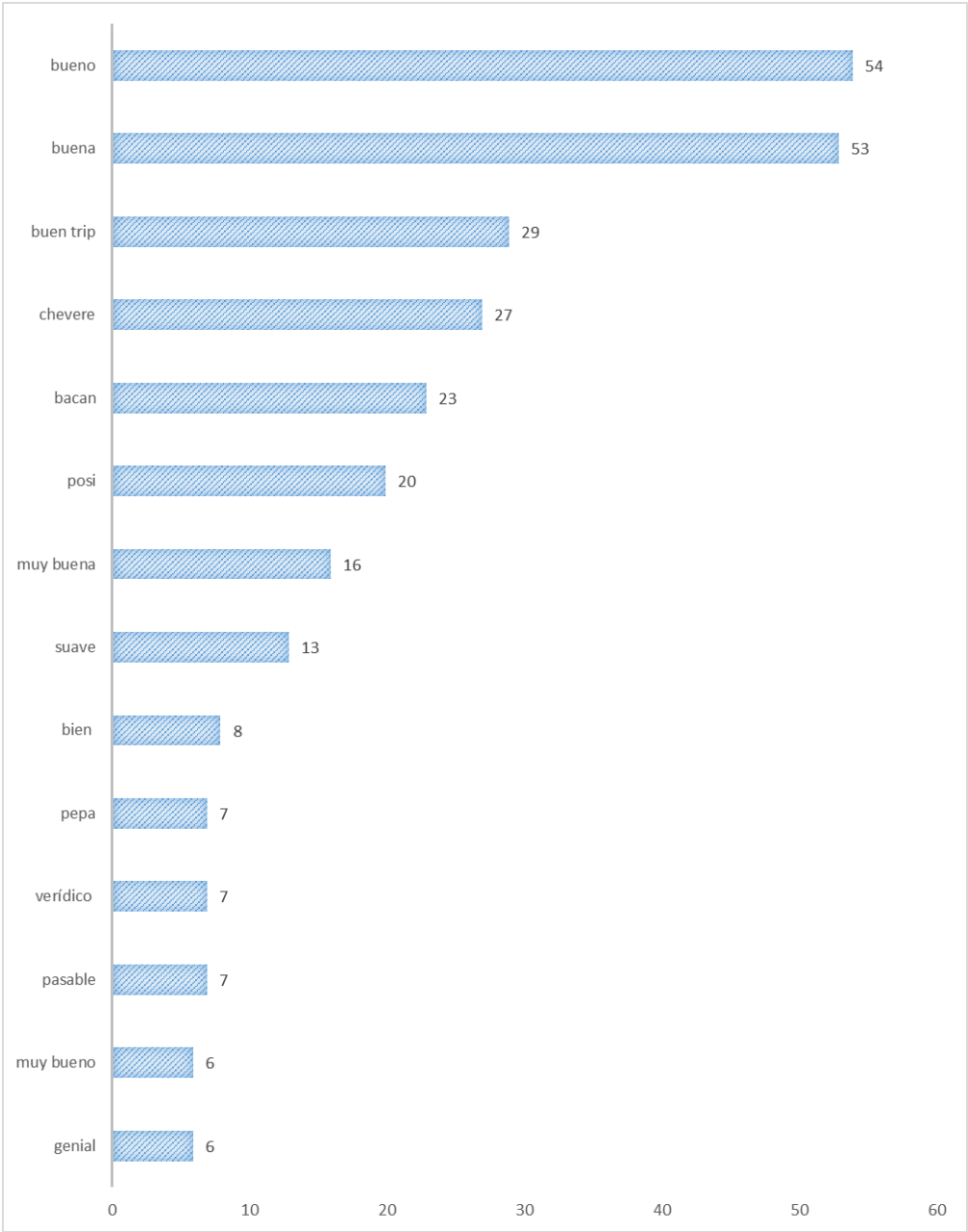


Figura 6. Expresiones Altamente Positivas más utilizadas. Elaborado por Autor.

La *Figura 6* presenta la relación entre el recuento de las expresiones más utilizadas para describir de la mejor manera (Altamente Positivo) a una persona en institución, según la visión de los usuarios digitales del País, quienes fueron encuestados a través de formularios de Google Forms. Mismas que formarán el listado de palabras con un valor de sentimiento de 5 según la escala de Likert, mencionada anteriormente.



*Figura 7. Expresiones Positivas más utilizadas. Elaborado por Autor.*

La *Figura 7* presenta la relación entre el recuento de las expresiones más utilizadas para describir una manera moderadamente buena (Positivo) a una persona en institución, según la visión de los usuarios digitales del País, quienes fueron encuestados a través de formularios de

Google Forms. Mismas que formarán el listado de palabras con un valor de sentimiento de 4 según la escala de Likert, mencionada anteriormente.

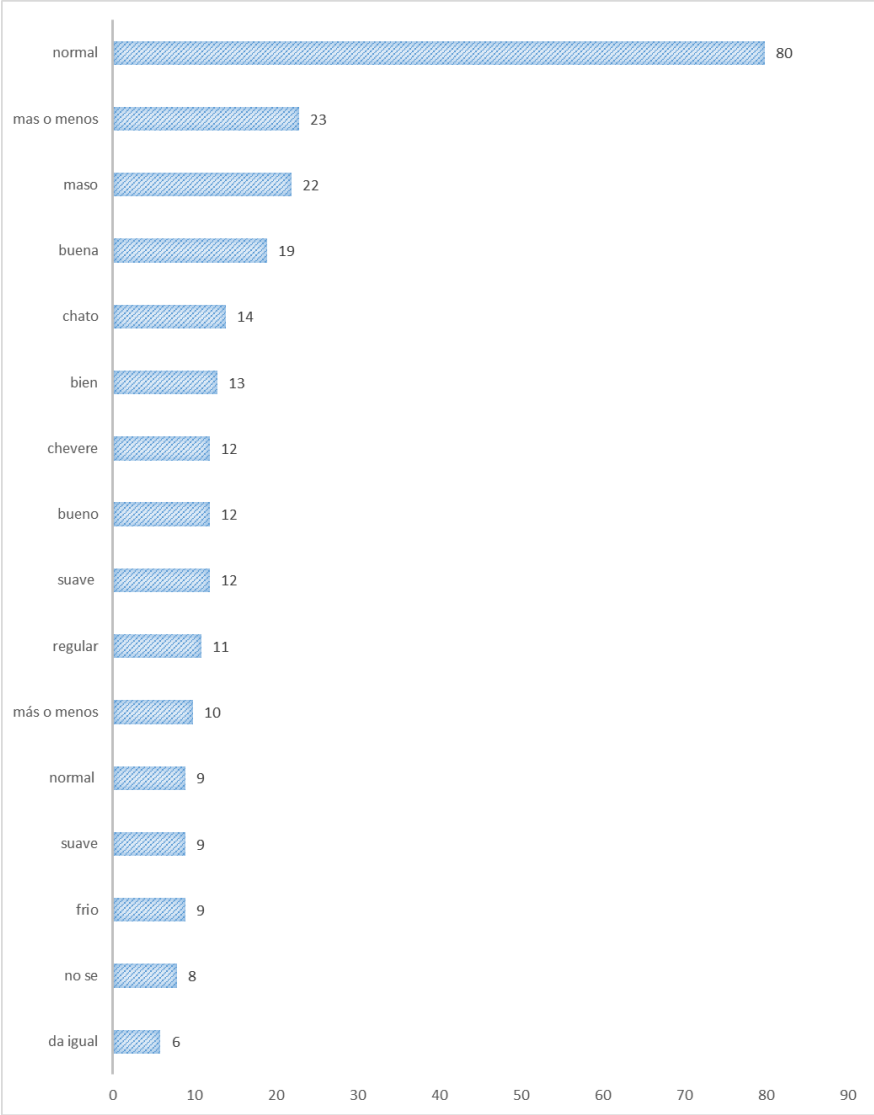
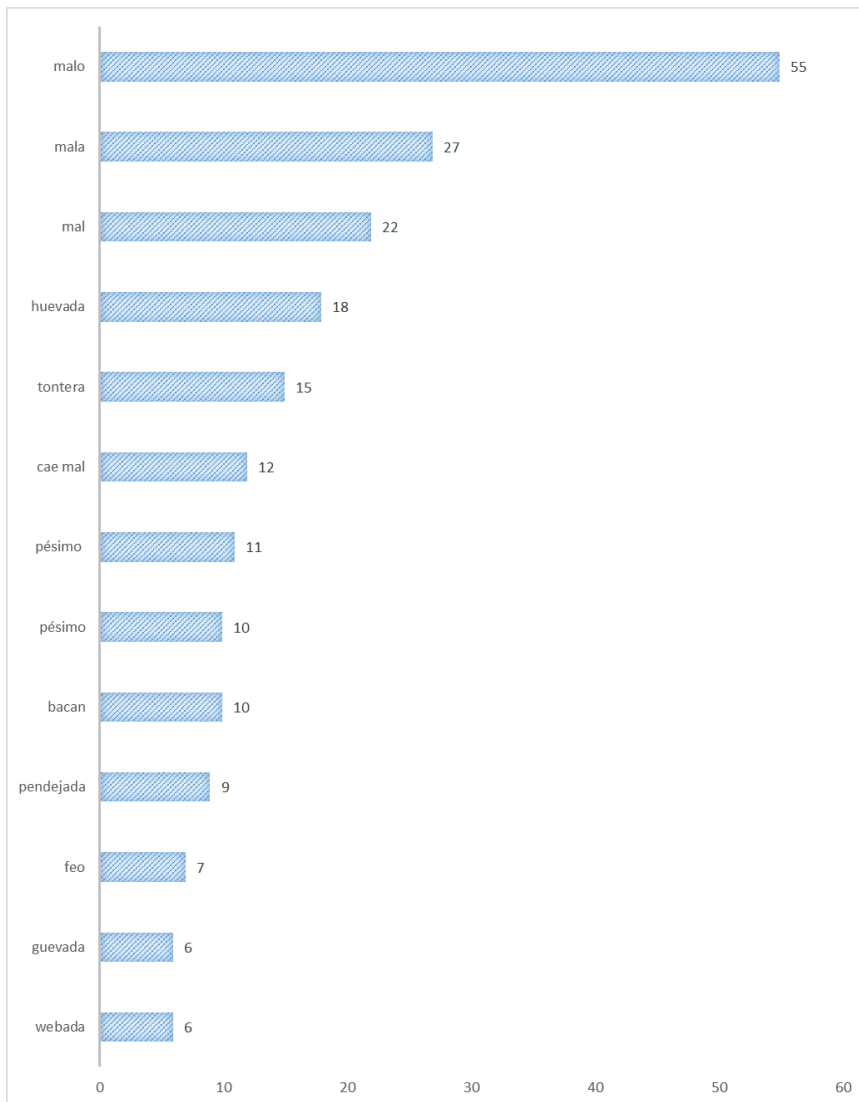


Figura 8. Expresiones Neutras más utilizadas Elaborado por Autor.

La Figura 8 presenta la relación entre el recuento de las expresiones más utilizadas para describir una manera indiferente (Neutral) a una persona en institución, según la visión de los usuarios digitales del País, quienes fueron encuestados a través de formularios de Google Forms. Mismas que formarán el listado de palabras con un valor de sentimiento de 3 según la escala de Likert, mencionada anteriormente.



*Figura 9. Expresiones Negativas más utilizadas Elaborado por Autor.*

La *Figura 9* presenta la relación entre el recuento de las expresiones más utilizadas para describir una manera moderadamente negativa (Negativa) a una persona en institución, según la visión de los usuarios digitales del País, quienes fueron encuestados a través de formularios de Google Forms. Mismas que formarán el listado de palabras con un valor de sentimiento de 2 según la escala de Likert, mencionada anteriormente.



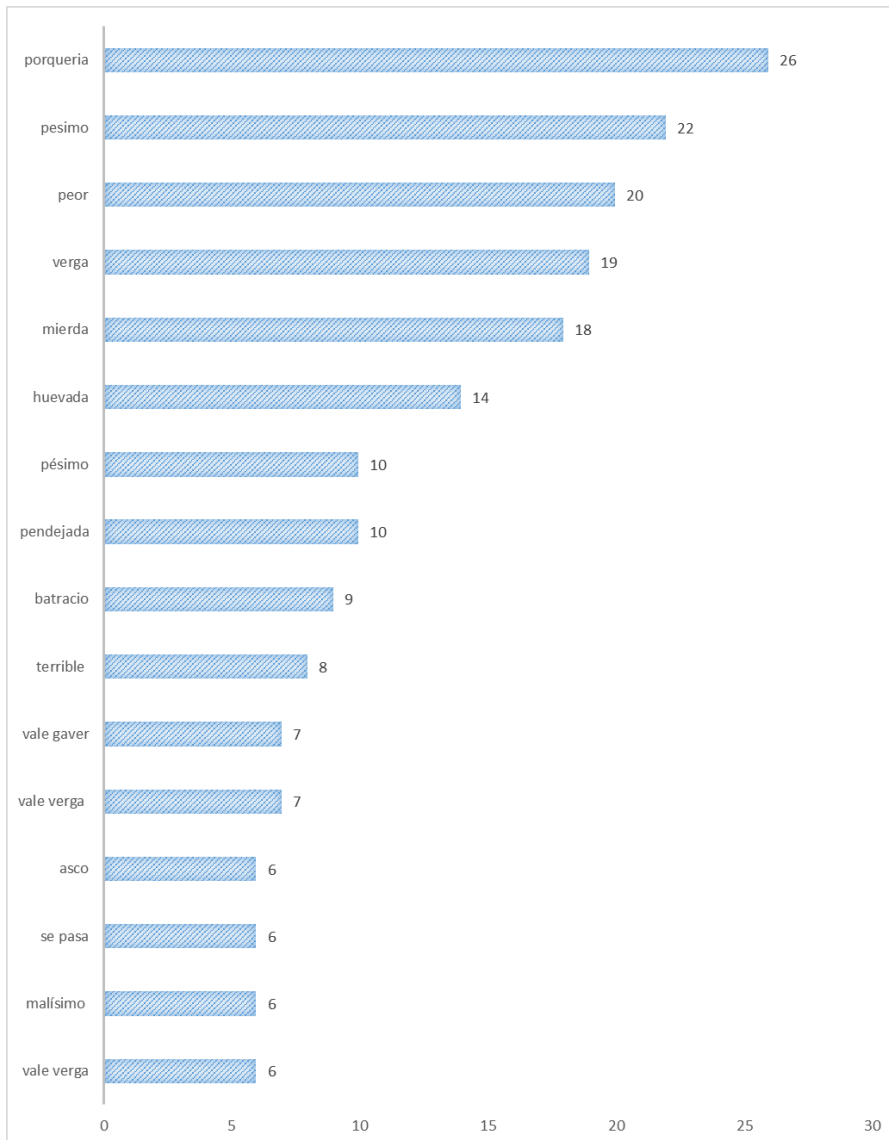


Figura 10. Expresiones Negativas más utilizadas Elaborado por Autor.

La *Figura 10* presenta la relación entre el recuento de las expresiones más utilizadas para describir una manera absolutamente negativa (Altamente Negativa) a una persona en institución, según la visión de los usuarios digitales del País, quienes fueron encuestados a través de formularios de Google Forms. Mismas que formarán el listado de palabras con un valor de sentimiento de 1 según la escala de Likert, mencionada anteriormente.

### 3.5 Análisis de Observación Directa

Como resultado de la observación directa realizada sobre redes sociales en opiniones y comentarios de los usuarios en temas controversiales, como política y deportes y se obtuvo un listado que de la misma manera que los resultados de las encuestas sigue la estructura de la escala de Likert, propuesta desde el inicio.

**Tabla 4.** Resultados observación directa para expresiones neutras y positivas

<b>Altamente positivas</b>	<b>Positivas</b>	<b>Neutras</b>
cheverisimo	posi	normal
bacansisimo	positivo	nada
bakansisimo	positiva	lo mismo
excelente	bacán	nada nuevo
maximo	bacano	
pepísima	bacana	
super	bacanada	
genial	bakan	
increíble	bakano	
del puctas	bakanada	
buenazo	chepo	
buenaso	pepa	
buenisimo	propio	
el éxito	cool	
el éxito		
la joda		
epico		
épico		
la pinta		
chévere		
chevere		

Elaborado por: Autor

**Tabla 5.** Resultados observación directa para expresiones negativas.

<b>Negativas</b>	<b>Altamente Negativas</b>
malo	puta
negativo	webada
ignorante	verga
pelucon	huevada
cholo	webada
negro	porqueria
lento	basura
tonto	borrego
estupido	perro
arrogante	zorro
vendido	burropositor
vago	burropositores
ricos	descerebrado
ricachon	ladron
pobre	tarrinero
pichagua	tarrinera
pinchagua	ladrona
bobo	corrupto
trozo	chucha
bolsa	choro
hueso	retrasado
pichita	corrupta
picha	miserable
pichuela	rata
	pelador
	sable
	vale pistola
	vales pistola

Elaborado por: Autor

En las *Tablas 4 y 5* se puede visualizar las expresiones registradas tras la observación directa realizada a través de la red social twitter a diferentes personas e instituciones, como: Rafael Correa (@MashiRafael) Guillermo Lasso, @LassoGuillermo, Barcelona Sporting Club (@BarcelonaSCweb) y Club Sport Emelec (@CSEmelec) y Pizza Hut Ecuador (@PizzaHutEC), cuyas interacciones constan de un volumen representativo para la recopilación de información.

### 3.6 Consolidado de Información

Como fruto de la consolidación de los resultados de las encuestas y la observación directa se obtuvo un listado de expresiones idiomáticas con su valor de sentimiento que fueron almacenadas en un archivo de texto para su lectura como léxico del prototipo funcional.

```
1 | chevere;5
2 | super;5
3 | bacano ;5
4 | buen trip;5
5 | placer;5
6 | muy bueno;5
7 | cheveruco;5
8 | bacan;5
9 | excelente;5
10 | excelente;5
11 | belleza;5
12 | chéverísimo;5
13 | guayaco;5
14 | chevere;5
15 | bacan;5
```

*Figura 11. Estructura archivo de léxicos. Elaborado por Autor*

El archivo almacenado responde a la estructura presentada en la ilustración 5, un archivo de texto separado por comas manteniendo la estructura en su primera posición (0) la expresión idiomática, y en la segunda posición (1) el valor de sentimiento.

## CAPÍTULO 4: PROPUESTA

Para medir la incidencia que tienen las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimiento, es necesario implementar el algoritmo en un prototipo de aplicación que permita visualizar y comparar los resultados previos a añadir soporte para las expresiones idiomáticas recopiladas en la en el capítulo anterior.

### 4.1 Viabilidad Técnica

Se requiere un lenguaje de programación robusto y de alto rendimiento para implementar un algoritmo de Análisis de Sentimiento, para el caso de este estudio se ha optado por **Python en su versión 3** por el motivo de su simplicidad sintáctica y su comprobada robustez, como fue descrito en el capítulo 2.4.1. Al enfrentar un problema de clasificación se requiere una librería que brinde funcionalidades de entrenamiento y evaluación de algoritmos de clasificación como **NLTK – Natural Language Toolkit**. También es necesario un serializador de objetos, como **Pickle**, para poder almacenar en disco los clasificadores entrenados y no ejecutar el proceso cada vez que la aplicación sea llamada. Para poder graficar en líneas de tendencia y hacer evidente la incidencia de agregar las expresiones idiomáticas propias de la jerga ecuatoriana se requiere una librería gráfica por lo que se optó por **matplotlib** la librería de graficación más usada en el ámbito científico en Python. Finalmente para realizar pruebas sobre información real se requiere obtener tweets referente a personas o temas específicos por lo que también será requerida la librería **Request** que permitirá hacer consultas HTTPS al API de Twitter.

Todos los elementos mencionados, que son requeridos para la implementación del prototipo de Análisis de Sentimiento son de código abierto y de licencia de uso libre, disponibles para su descarga, instalación y uso, por lo que no se incurrirá en gastos financieros para implementar el mismo.

### 4.2 Descripción del Software

#### 4.2.1 Python 3

El lenguaje de programación Python, fue elegido para el desarrollo del prototipo funcional por su sintaxis sencilla, su gran comunidad en línea y el poder de sus librerías para clasificación. Estas características fueron mencionadas en el capítulo 2.4.1 y permitirán realizar la implementación en el tiempo requerido por el proyecto de investigación.

Su sintaxis sencilla brinda como ventajas: la disminución representativa en la pendiente de la curva de aprendizaje y a la vez codificación rápida y eficaz al usar menos líneas de código.

La existencia de una gran comunidad de desarrolladores Python genera confianza al momento de optar por un lenguaje de desarrollo, ya que la misma implica la existencia de soporte y documentación de errores comunes. Finalmente el poder de sus librerías facilitará la implementación de algoritmos de Aprendizaje Automático

## **4.2.2 NLTK**

Esta librería posee un conjunto de herramientas para el procesamiento de lenguaje natural, y Aprendizaje Automático que permitirán desarrollar funciones de alta importancia en el proceso de Análisis de Sentimiento de manera rápida.

### **4.2.2.1 Disgregación de las Palabras del Documento (Tokenization)**

Ya que el enfoque de Análisis de Sentimiento a utilizarse en el proyecto será **basado en léxicos** se requiere disgregar las palabras del texto a analizarse para ser clasificadas una a una. Para este proceso NLTK cuenta con funciones de disgregación y un módulo corpus que permitirá reconocer palabras sin contenido de valor sentimental como artículos, adverbios y signos de puntuación.

#### **4.2.2.2 Léxicos**

Los léxicos son archivos de texto, de estructura variable pero que para el objetivo de este estudio deben poseer, necesariamente, dos datos: la expresión idiomática y el sentimiento. Estos pueden ser archivos de valores separados por coma (CSV), o de valores separados por tabulaciones (TSB), entre otros... Estos deben ser cargados y procesados por el prototipo para entrenar al Clasificador Bayesiano. En el caso de los léxicos que poseen expresiones multipalabras se reemplaza los espacios en blanco por guion bajo para que el clasificador pueda tratarlos como una característica única.

#### **4.2.2.3 Clasificador Ingenuo Bayesiano**

Para realizar la categorización de las expresiones será necesario un clasificador que sea entrenado y puesto a prueba con los léxicos mencionados en el apartado anterior. **NaiveBayesClassifier** es una clase del paquete NLTK, cuya instancia es un clasificador Naive Bayes, algoritmo de aprendizaje supervisado descrito en capítulo 2.4.3.

## **4.2.3 Serializador de Objetos**

Pickle es una librería que permite serializar objetos, permitiendo almacenar los mismos en archivos que pueden ser recuperados por la misma librería. La utilidad de esta librería para

el prototipo es la necesidad de almacenar los clasificadores bayesianos para evitar ejecutar el proceso de entrenamiento cada vez que se ejecuta la aplicación.

#### 4.2.4 Requests

Este paquete permite realizar consultas a través del protocolo de transferencia segura de hipertexto (HTTPS). Para establecer una conexión con los servicios de Twitter es necesario realizar consultas a través de su API, usando un identificador provisto por la misma página para obtener un listado de tweets.

### 4.3 Detalles de la Propuesta

Como proyecto de código abierto, el prototipo debe tener un nombre que lo distinga de los demás alojados en Github, por lo que el nombre será SENTEC correspondiendo a “Sentiment Ecuador”, nombre en inglés para su uso internacional como modelo de Análisis de Sentimiento basado en léxicos.

#### 4.3.1 Estructura de carpetas

Sentec es una aplicación sencilla y modular, desarrollada en Python, por lo que corresponde a una estructura básica para aislar lógica de programación (Módulos) y recursos.

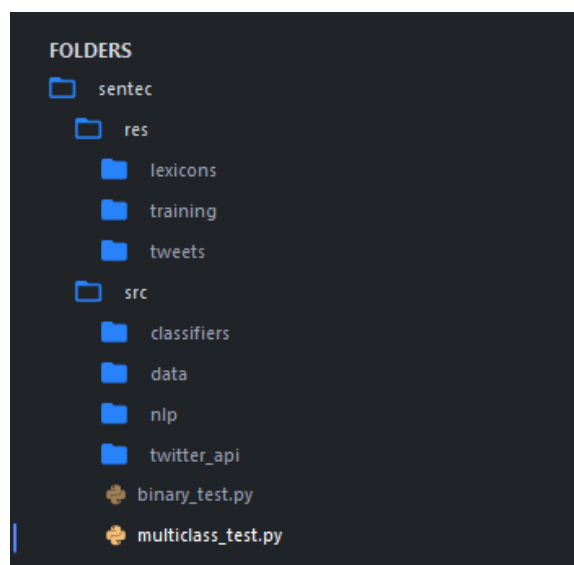


Figura 12. Estructura de Carpetas. Elaborado por Autor.

##### 4.3.1.1 Carpeta de Recursos

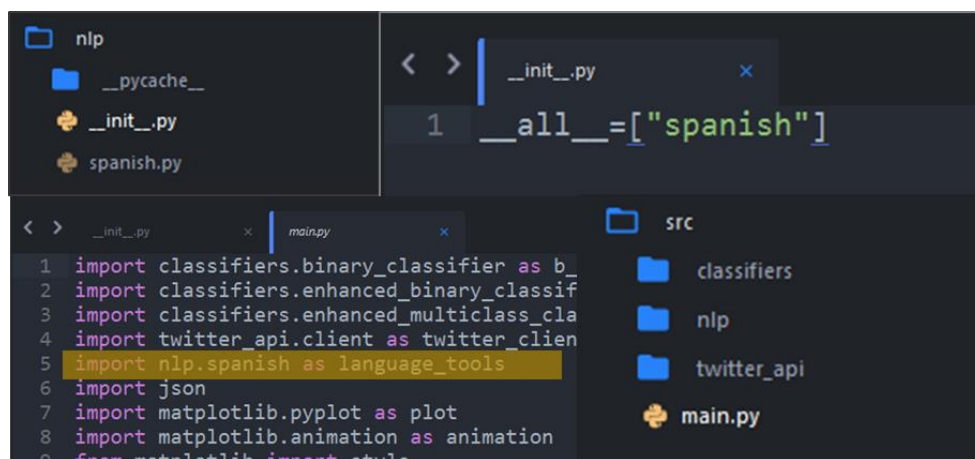
La carpeta de “res” corresponde a los recursos de utilizados o generados por los diferentes scripts de la aplicación. La primera “lexicons” contiene los léxicos que serán utilizados para entrenar los clasificadores. La carpeta “training”, archivos de texto con datos de los clasificadores bayesiano serializados por la librería **pickle**. Finalmente la carpeta “tweets”,

archivos JSON con listas de tweets generadas por script de captura de tweets. También en la carpeta tweets, con el objetivo de realizar una prueba guiada se añadió un archivo con texto con resultados conocidos, al hacer pruebas sobre estas opiniones los resultados y la graficación tendrán una tendencia controlada conocida, permitiendo visualizar la diferencia entre los algoritmos.

#### 4.3.1.2 Carpeta de Fuentes

El código fuente de sentec se encuentra en la carpeta “src” (Source) que contiene el archivo principal para ejecutar aplicación “main.py” y las carpetas de los módulos desarrollados: la carpeta “classifiers” contiene el código de los clasificadores bayesianos; la carpeta “nlp” contiene el módulo de Procesamiento de Lenguaje Natural con funciones mínimas desarrolladas para el idioma español; la carpeta “twitter\_api” con los scripts que permiten extraer tweets de temáticas generales del API de Twitter; la carpeta “data” posee el modulo “resources” que permite acceder a los recursos guardados como los “tweets”.

Las carpetas \_\_pycache\_\_ son generadas por Python para fines de cacheo, como lo indica su nombre descriptivo. Los archivos \_\_init\_\_.py permiten a los modulos compartir sus archivos y ponerlos en el mismo espacio de nombre que los demás para poder ser importados en cualquier archivo del proyecto.



The screenshot shows a code editor with two windows. The top window is titled “\_\_init\_\_.py” and contains the following code: 

```
1 __all__=["spanish"]
```

 The bottom window is titled “main.py” and contains the following code: 

```
1 import classifiers.binary_classifier as b_
2 import classifiers.enhanced_binary_classif
3 import classifiers.enhanced_multiclass_cla
4 import twitter_api.client as twitter_clien
5 import nlp.spanish as language_tools
6 import json
7 import matplotlib.pyplot as plot
8 import matplotlib.animation as animation
9 from matplotlib import style
```

 On the right side of the editor, a file explorer shows the directory structure: `src` (folder), `classifiers` (folder), `nlp` (folder), `twitter_api` (folder), and `main.py` (file).

Figura 13. Detalles archivo de inicialización e importación python. Elaborado por Autor.



## 4.3.2 Funcionalidades de librerías creadas

A continuación se describe las funcionalidades diseñadas para realizar análisis de sentimiento basado en léxicos, para el caso particular de este proyecto. La aplicación se ejecuta en 2 etapas que corresponden a scripts independientes que serán descritas a continuación.

### 4.3.2.1 Obtención de Tweets

La primera etapa corresponde al script de obtención de tweets que nos permite descargar tweets usando el API REST de Twitter, enviando un parámetro para la consulta, que puede ser el usuario de twitter de una persona, institución o incluso tendencias. Un paso previo para poder consultar el API de twitter es registrar una aplicación en el sitio de desarrolladores de twitter y crear una clave de accesos (Access Token). A continuación se describe el algoritmo en pseudocódigo de la primera etapa.

```
OBTENCIÓN DE TWEETS
> Entrada: Usuario de Twitter para consulta
> Envió de solicitud al API de Twitter
> Paginación de resultados
  Acumulación de resultados
  Si existen más paginas
    Agregar parámetros y volver al paso 2
> Guardar archivo en res/NOMBRE_USUARIO-tweets.json
```

Figura 14. Pseudocódigo obtención y paginación de Tweets. Elaborado por Autor.

```
1 # twitterApi - Twitter Rest API Simplified Client
2 # author: Julio Vasquez
3
4 import json
5 import requests
6 from operator import itemgetter
7 access_token="AAAAAAAAAAAAAAAAAAAAH6RxA4AAAA%2F%2FhkyD1b8mISa1JH%2Bpb%2FBzWFKI4%3DZzGw8dlYzgc4dvdqotvtMb"
8 iteration_limit=100
9 iteration_count=0
10 def getTweets(params):
11     global iteration_count
12     print("Iteracion "+ str(iteration_count))
13     print("Obteniendo Tweets Referentes a "+params["q"]+"...")
14     headers={"Authorization":"Bearer "+access_token}
15     result = requests.get("https://api.twitter.com/1.1/search/tweets.json",params=params,headers=headers)
16     json_result=json.loads(str(result.text))
17     return json_result
18
19
20 def getAllTweets(params,lowest_id=0,all_statuses=[]):
21     global iteration_count
22     global iteration_limit
23     iteration_count+=1
24     if(lowest_id!=0):
25         params["max_id"]=lowest_id-1
26
27     new_response=getTweets(params)
28
29     if(len(new_response["statuses"])==0) or iteration_count>=iteration_limit:
30         return new_response["statuses"]
31     else:
32         statuses=new_response["statuses"]
33         lowest_id=min(statuses,key=itemgetter("id"))
34         return statuses+getAllTweets(params,lowest_id=lowest_id["id"])
35
36
37
```

Figura 15. Captura de código del módulo de obtención de Tweets. Elaborado por Autor.

### 4.3.2.2 Análisis de Sentimiento

El script principal de la aplicación main.py ejecuta todos los procesos nucleares del sistema, se ha optado por separar las funcionalidades para permitir su funcionamiento fuera de línea. Este es un macro proceso está compuesto por un conjunto de procesos de alta importancia por lo que serán descritos a continuación a detalle.

#### 4.3.2.2.1 Clasificadores Bayesianos

Antes de poder realizar análisis de sentimiento, fue necesario crear y entrenar los clasificadores bayesianos. Mismos que permiten realizar la clasificación de las expresiones idiomáticas con las que sean entrenados.

```
Entrenar Clasificador Bayesiano
> Obtener archivo de léxicos
> Extraer palabras y valor de sentimiento
> Crear arreglo para entrenamiento
> Instanciar el Objeto NaiveBayesClassifier
> Entrenar el objeto con los datos del léxico
```

Figura 16. Pseudocódigo del módulo de entrenamiento de un Clasificador Bayesiano.

```
1 import nltk
2 import pickle
3 import os
4 from nlp.spanish import stem
5
6 def extract_feature(w):
7     return {"word":w}
8
9 lexicon_lines=open("../res/lexicons/binary_lexicon.txt","r", encoding='utf-8').readlines();
10 classifier_file_name="../res/training/binary_classifier.pickle"
11
12 labeled_words=[]
13 word_list=[]
14
15 for line in lexicon_lines:
16     split=line.replace("\n","").split("\t");
17     word=stem(split[0].strip().replace(" ","_"))
18     word_list.append(word);
19     labeled_words.append((word,split[2]))
20 #construct features
21 feature_set=[extract_feature(n), sentiment) for (n, sentiment) in labeled_words]
22
23 def getClassifier():
24     if not (os.path.isfile(classifier_file_name)):
25         classifier = nltk.NaiveBayesClassifier.train(feature_set)
26         f = open(classifier_file_name, 'wb')
27         pickle.dump(classifier, f)
28         f.close();
29     else:
30         f = open(classifier_file_name, 'rb')
31         classifier = pickle.load(f)
32         f.close()
33     return classifier
34
35 def getWordList():
36     return word_list
37
38 def getFeatureSet():
39     return feature_set
```

Figura 17. Captura del código de un clasificador Binario. Elaborado por Autor.

Los clasificadores que realizan la clasificación entre positivo y negativo, son **binarios** ya que poseen solamente dos clases para realizar la clasificación. Por otro lado los que tienen la capacidad de clasificar entre diferentes clases son clasificadores **multiclase**.

#### 4.3.2.2.2 Disgregación de las palabras y obtención de las raíces

Son dos procesos que extienden del procesamiento de lenguaje natural, conocidos como “Tokenization” y “Stemming” en inglés. El proceso de disgregación de las palabras tiene como objetivo separar las palabras del texto ingresado, esto es necesario ya que el proceso de Análisis de Sentimiento fue **enfocado en léxicos**, en las expresiones idiomáticas. El proceso de obtención de las raíces consiste en extraer la raíz de las palabras usadas, esto con el fin de evitar que las conjugaciones afecten el análisis. Finalmente **para poder procesar expresiones con más de una palabra** como “muy bueno”, “buen trip”, es necesario formar expresiones **para incrementar los alcances del análisis**.

```
1 import nltk
2 import scipy
3 from nltk.corpus import stopwords
4 from nltk import word_tokenize
5 from nltk.stem import SnowballStemmer
6 from string import punctuation
7
8 spanish_stopwords = stopwords.words('spanish')
9 stemmer = SnowballStemmer('spanish')
10
11 non_words = list(punctuation)
12 non_words.extend(['¿', '¡'])
13 non_words.extend(map(str, range(10)))
14
15 def stem(word):
16     return stemmer.stem(word)
17
18 def stem_tokens(tokens):
19     stemmed = []
20     for item in tokens:
21         stemmed.append(stem(item))
22     return stemmed
23
24 def tokenize(text):
25     text = ''.join([c for c in text if c not in non_words])
26     tokens = word_tokenize(text)
27     tokens = stem_tokens(tokens)
28     bigrams=[ x.lower()+"_"+y.lower() for (x,y) in zip(tokens,tokens[1:])]
29     trigrams=[ x.lower()+"_"+y.lower() for (x,y) in zip(bigrams,tokens[2:])]
30     tokens=[t.lower() for t in tokens ]+bigrams+trigrams
31     return tokens
32
33
```

Figura 18. Captura del módulo de Procesamiento de Lenguaje Natural para el idioma Español. Elaborado por Autor.

El módulo **nlp.spanish** posee las funcionalidades para realizar las funciones mencionadas de manera aislada. De esta forma ha permitido procesar las expresiones presentes en los léxicos y en los cuerpos de texto a procesarse previo a su análisis.

### 4.3.3 Construcción del Prototipo

Para la construcción del prototipo se requiere un punto de partida para la ejecución de las pruebas pertinentes de las librerías implementadas. Los scripts principales se encuentran en la raíz de la carpeta “src” el objetivo de cada script es el de realizar las evaluaciones correspondientes de los diferentes algoritmos de clasificación.

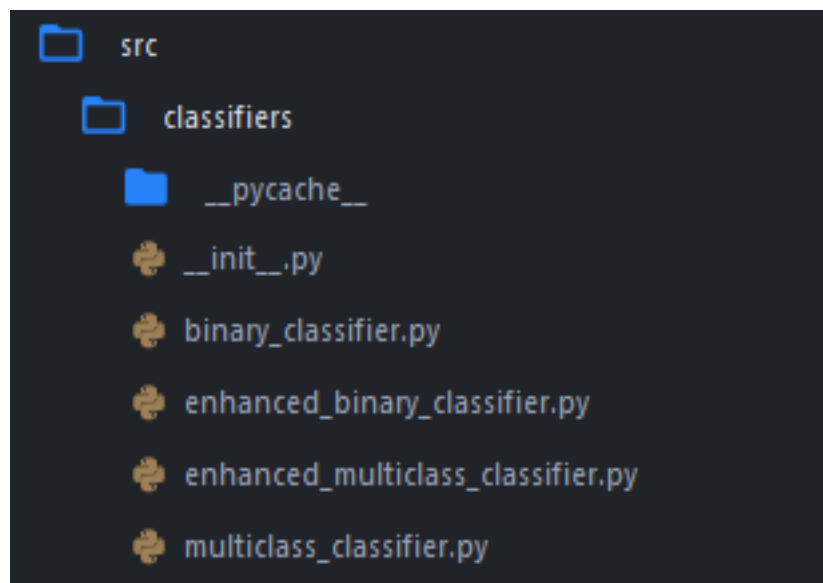


Figura 19. Estructura de carpeta de código fuente. Elaborado por Autor.

Se implementaron cuatro clasificadores de dos tipos diferentes: los clasificadores Binarios (Positivo – Negativo) y los clasificadores Multiclase (del 1 al 5). Con el fin de comparar la precisión del mismo algoritmo al añadir en su lista de características de entrenamiento el léxico ecuatoriano.

**Tabla 6.** Descripción de los módulos de clasificadores.

<b>Clasificador</b>	<b>Características</b>
binary_classifier.py	Clasificador Binario <b>Léxico:</b> Sentiment Lexicons in Spanish by Veronica Perez-Rosas, Carmen Banea and Rada Mihalcea
enhanced_binary_classifier.py	Clasificador Binario <b>Léxico:</b> <ul style="list-style-type: none"><li>• Sentiment Lexicons in Spanish by Veronica Perez-Rosas, Carmen Banea and Rada Mihalcea</li><li>• Léxico Ecuador: Resultado de las Encuestas</li></ul>
multiclass_classifier.py	Clasificador Multiclase <b>Léxico:</b> <ul style="list-style-type: none"><li>• Sentiment Lexicons in Spanish by Veronica Perez-Rosas, Carmen Banea and Rada Mihalcea</li></ul>
enhanced_multiclass_classifier.py	Clasificador Multiclase <b>Léxico:</b> <ul style="list-style-type: none"><li>• Sentiment Lexicons in Spanish by Veronica Perez-Rosas, Carmen Banea and Rada Mihalcea</li><li>• Léxico Ecuador: Resultado de las Encuestas</li></ul>

Elaborado por: Autor

Como recurso de ingreso para la el prototipo se tiene varias listas de tweets generadas por el módulo de obtención de tweets, añadido a esto un archivo “test.txt” con la lista de opiniones de prueba, mencionada en el capítulo 4.2.1.1.

Al procesar los tweets se obtiene una lista de textos (opiniones), las mismas luego deben pasar por el proceso de disgregación de las expresiones, de manera que antes de clasificar se tenga una lista de expresiones idiomáticas extraídas de la opinión. Los resultados de la clasificación se acumulan y se cuenta las veces que las palabras son clasificables, con el fin de poder comparar la precisión de los clasificadores con sus versiones mejoradas. Estos datos serán presentados en la consola y en una gráfica de líneas de sentimiento en la que será posible contemplar y comparar los resultados de la clasificación de los diferentes algoritmos.

```

14 def classifyTweets(classifier_module,tweets,plt,line_color,name):
15     classifier=classifier_module.getClassifier()
16     x=0
17     y=0
18     xar=[]
19     yar=[]
20     word_list=classifier_module.getWordList()
21     for t in tweets:
22         tokens=language_tools.tokenize(t)
23         sentiment_word_count=0
24         sum_sentiment=0
25         for word in tokens:
26             if word in word_list:
27                 classified_words[name] += 1
28                 sentiment_word_count += 1
29                 if classifier.classify(classifier_module.extract_feature(word))=="pos":
30                     sentiment_sum[name] += 1
31                     y+=1
32                 else:
33                     y-=1
34
35             x += 1
36             xar.append(x)
37             yar.append(y)
38     plt.plot(xar,yar,line_color)
39

```

Figura 20. Captura del código de la función de clasificación de texto. Elaborado por Autor.

El método **classifyTweets** realiza los procesos mencionados, procesando el texto y luego clasificando las palabras, aumentando los acumuladores y agregando los resultados de clasificación al grafico.

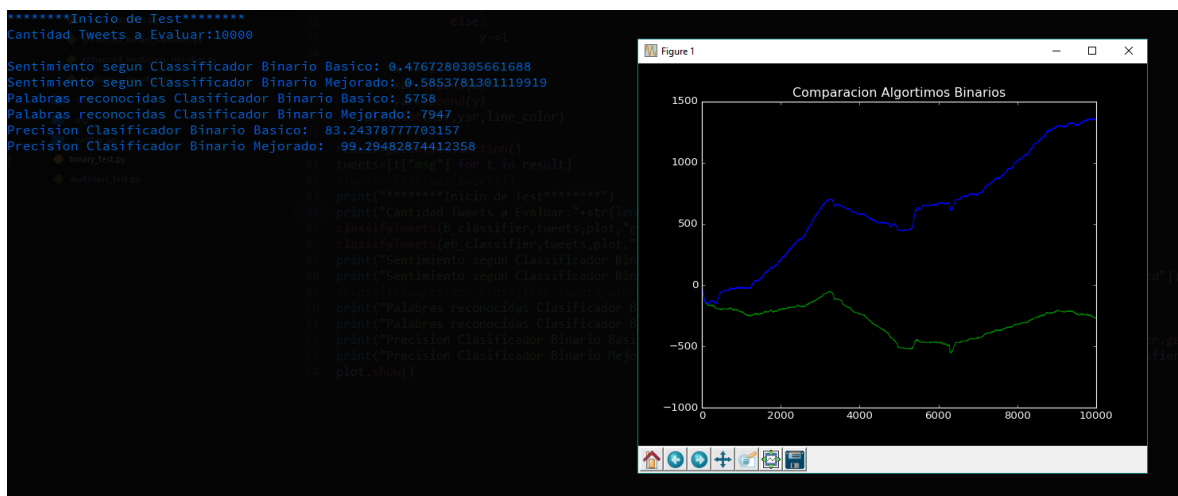


Figura 21. Captura de resultados de ejecución de prototipo. Elaborado por Autor.

Los resultados de la ejecución tienen la apariencia presentada en la *Figura 21* y serán analizados en el siguiente capítulo.

### 4.3.4 Diagrama de flujo de Análisis de Sentimiento

La representación visual de los procesos y subprocesos que conlleva el desarrollo de la minería de Opiniones, cada uno de los procesos que ha sido mencionado en los apartados 4.3.2 y 4.3.3 responden a un orden para realizar correctamente el proceso de análisis de sentimiento.

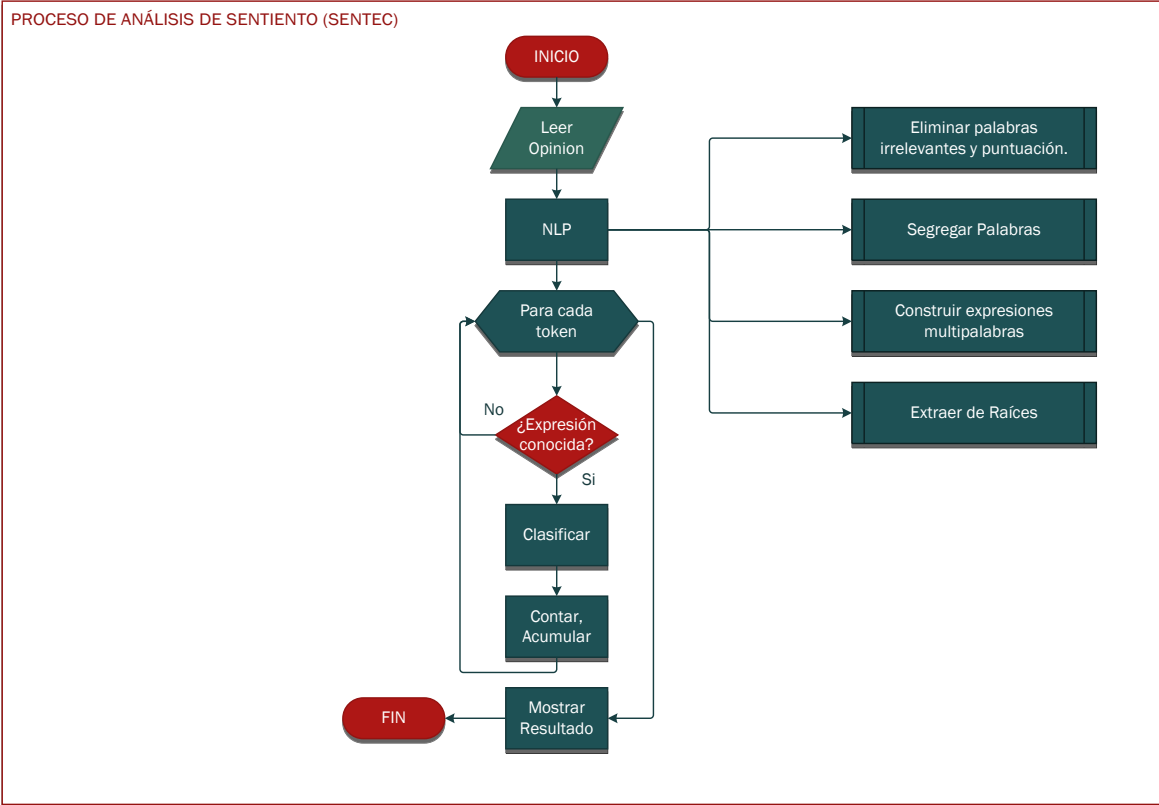


Figura 22: Diagrama de Procesos, Elaborado por: Autor

### 4.4 Análisis de Resultados

Los resultados de diversas ejecuciones serán presentados y comparados a continuación para describir la incidencia de agregar expresiones idiomáticas de la jerga ecuatoriana a los algoritmos de análisis de sentimiento.

```
*****Inicio de Test*****
Cantidad Tweets a Evaluar:111
Sentimiento segun Clasificador Binario Basico: 0.5
Sentimiento segun Clasificador Binario Mejorado: 0.5777777777777777
Palabras reconocidas Clasificador Binario Basico: 88
Palabras reconocidas Clasificador Binario Mejorado: 180
Precision Clasificador Binario Basico: 83.24378777703157
Precision Clasificador Binario Mejorado: 99.29482874412358
```

Figura 23. Resultados en consola de la ejecución del prototipo. Elaborado por Autor.

En la *Figura 23* se contempla los resultados de la ejecución del script de pruebas de clasificadores binarios “binary\_test.py”, en la que se puede contemplar los resultados de ambos clasificadores binarios, el básico y el mejorado, cuyas diferencias fueron explicadas en el capítulo 4.3.3. Visualizando los resultados de la ejecución **se evidencia la incidencia que ha tenido agregar al entrenamiento del algoritmo de clasificación las expresiones idiomáticas ecuatorianas** al comparar los resultados de cada clasificador.

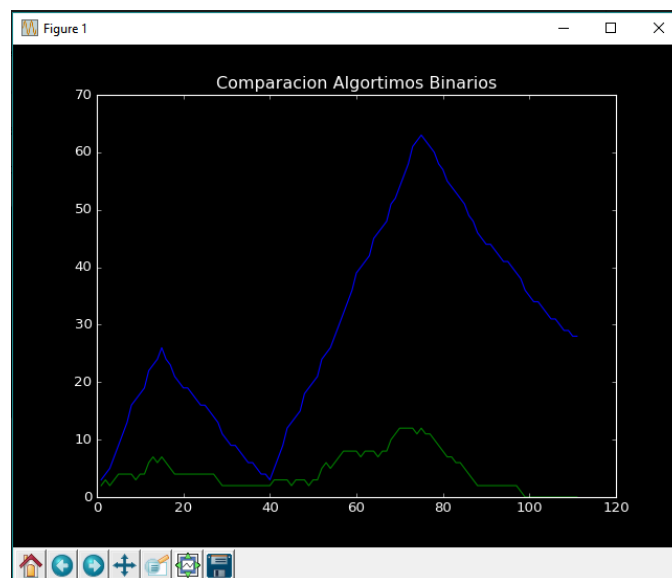


Figura 24. Visualizaciones de tendencia de Sentimiento generadas por el prototipo. Elaborado por Autor.

También al finalizar la ejecución se presenta una gráfica con dos líneas, que representan la tendencia positiva o negativa de las expresiones en el eje Y, y el número de la frase analizada en el eje X. La línea verde describe la tendencia de sentimiento clasificada por el algoritmo básico. La línea azul describe la tendencia de sentimiento clasificada por el algoritmo extendido o mejorado. Esta gráfica nos permite visualizar la diferencia entre ambos clasificadores que toman tendencias.



La misma situación se repite para el clasificador Multiclase que describe una tendencia bastante similar al clasificador binario, pero agregando un sentido de intensidad a las palabras positivas y negativas, influyendo con 5 niveles de sentimiento.

```
*****Inicio de Test*****
Cantidad Tweets a Evaluar:111

Sentimiento segun Classificador Binario Basico: 0.5
Sentimiento segun Classificador Binario Mejorado: 0.5777777777777777
Palabras reconocidas Clasificador Binario Basico: 88
Palabras reconocidas Clasificador Binario Mejorado: 180
Precision Clasificador Binario Basico contra Mejorado: 66.66666666666666
```

Figura 25. Captura de resultados de ejecución de clasificador multiclase. Elaborado por Autor.

Al comparar resultados la imprecisión del algoritmo básico se hace mucho más evidente, por el hecho de existir más de dos niveles de sentimiento, siendo esto un limitante entre los léxicos existentes para el idioma español, y específicamente del que se ha usado para la prueba, que clasifica las palabras de forma binaria, como positivas y negativas. Mientras los resultados de las encuestas y observación directa permitieron extender el alcance de la clasificación a 5 clases.

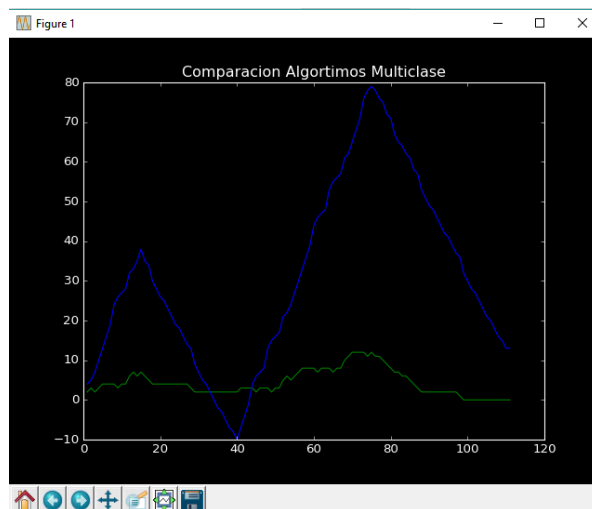


Figura 26. Captura de líneas de tendencia, resultados de clasificador multiclase. Elaborado por Autor.

En las líneas de tendencia presentadas, se puede contemplar el un resultado similar al de los algoritmos binarios, pero con tendencias más pronunciadas, tanto positiva como negativamente, esto permite aumentar la precisión al clasificar palabras con cargas emocionales más altas como el caso de “bueno” y “excelente”, “malo” y “porquería”.

Los resultados presentados hasta el momento corresponden a un paquete de opiniones construido para la prueba cumpliendo con la tendencia diseñada para las mismas. Del 1 al 15

positivas, del 16 al 40 negativas, del 41 hasta el 75 positivas, y del 76 al 110 negativas. A continuación se presentarán la tabulación de los resultados para recursos obtenidos de twitter y su comparativa.

**Tabla 7.** Cantidad de expresiones reconocidas.

Datos de clasificación		Clasificadores binarios			Clasificadores multiclase		
Conjunto de datos	Cantidad de Opiniones	Básico	Mejorado	$\Delta$	Básico	Mejorado	$\Delta$
test.txt	111	88	180	105%	88	188	114%
tweets- BarcelonaSCweb.json	1290	316	763	141%	316	858	172%
tweets-CSEmelec.json	1471	512	1097	114%	512	1144	123%
tweets-MashiRafael.json	9999	5973	8344	40%	5973	9257	55%
tweets-PizzaHutEC.json	79	44	96	118%	44	97	120%
<b>Promedio</b>				<b>104%</b>			<b>117%</b>

Elaborado por: Autor

La *Tabla 7* registra la cantidad de palabras reconocidas por los algoritmos de clasificación, una primera idea de la evolución del algoritmo básico al añadir en su entrenamiento las expresiones idiomáticas ecuatorianas. Se puede observar que la incidencia es altamente positiva al reconocer en promedio 104% más palabras para el algoritmo binario y 117% más palabras en el algoritmo multiclase.

**Tabla 8.** Aceptación según clasificadores.

DATOS DE CLASIFICACION	CLASIFICADORES BINARIOS				CLASIFICADORES MULTICLASE		
	Cantidad de Opiniones	Básico	Mejorado	$\Delta$	Básico	Mejorado	$\Delta$
test.txt	111	50,0%	57,8%	7,8%	60,0%	61,4%	1,4%
tweets-BarcelonaSCweb.json	1290	68,7%	79,9%	11,3%	67,5%	61,3%	6,2%
tweets-CSEmelec.json	1471	63,1%	46,8%	16,4%	65,3%	51,4%	13,9%
tweets-MashiRafael.json	9999	54,2%	64,7%	10,5%	61,7%	63,8%	2,1%
tweets-PizzaHutEC.json	79	59,1%	74,0%	14,9%	63,6%	67,8%	4,2%
Promedio				12,2%			5,6%

Elaborado por: Autor

La *Tabla 8* registra los niveles de aceptación de los diferentes clasificadores puestos a prueba. Se puede observar los resultados como porcentaje de aceptación, también cada tipo de clasificador posee una columna de diferencia, que hace evidente la incidencia que la extensión del léxico tiene sobre los resultados de la clasificación, afinando la salida que será presentada a los usuarios.

**Tabla 9.** Precisión de algoritmos básicos, comparados con los resultados de los algoritmos mejorados.

DATOS DE CLASIFICACIÓN		CLASIFICADORES BINARIOS		CLASIFICADORES MULTICLASE	
Conjunto de datos	Cantidad de Opiniones	Básico	Error	Básico	Error
test.txt	111	67%	33%	33%	67%
tweets-BarcelonaSCweb.json	1290	48%	52%	36%	64%
tweets-CSEmelec.json	1471	83%	17%	45%	55%
tweets-MashiRafael.json	9999	74%	26%	63%	37%
tweets-PizzaHutEC.json	79	53%	47%	46%	54%
<b>Promedio</b>	<b>2590</b>	<b>65%</b>	<b>35%</b>	<b>45%</b>	<b>55%</b>

Elaborado por: Autor

Finalmente en la *Tabla 9* se contempla los porcentajes de precisión y error de los algoritmos básicos, respecto a los mejorados. Estos niveles de precisión son resultado de evaluar el algoritmo clasificador básico contra los resultados del clasificador mejorado de su tipo respectivos, Esta verificación puede ser realizada gracias a la funcionalidad de precisión (accuracy) del paquete NLTK. Los resultados presentan niveles de error inaceptables como 35% y 55%, al compararse los algoritmos básicos contra los mejorados.

## CAPÍTULO 5: CONCLUSIONES

- 1) **La polaridad e intensidad** de las expresiones idiomáticas puede cambiar de acuerdo a la región geográfica, en sus diferentes niveles de granularidad: continente, región, país, e incluso ciudad o sector. **Agregar a los léxicos** no sólo **polaridad** sino **niveles de intensidad**, permite **sumar precisión** a los algoritmos de Análisis de Sentimiento;
- 2) En el enfoque basado en léxicos del Análisis de Sentimiento, las expresiones idiomáticas propias de una jerga ejercen un impacto importante en los resultados de la clasificación, como se ha podido evidenciar en el apartado 4.4, al incluir en el entrenamiento de los algoritmos las expresiones idiomáticas propias de la jerga ecuatoriana, **se presenta un aumento significativo la cantidad de palabras identificadas** por los algoritmos;
- 3) A un mayor volumen de expresiones idiomáticas reconocidas por el algoritmo de clasificación, le sigue el aumento en la precisión del algoritmo de Análisis de Sentimiento. Esta relación influye directamente en los resultados del análisis, como se pudo contemplar en el apartado 4.4;
- 4) El soporte de expresiones que contengan más de una palabra permite enriquecer el léxico de entrenamiento y, por ende, reduce el riesgo de ambigüedad en el Análisis de Sentimiento basado en léxicos al simular un contexto para las palabras a evaluarse.

## Bibliografía

- Collomb, A., Costea, C., Joyeux, D., Hasan, O., & Brunie, L. (s. f.). A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation. Recuperado a partir de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.646.9070&rep=rep1&type=pdf>
- Donzeau-Gouge, V., Huet, G., Kahn, G., & Lang, B. (1980). *Programming environments based on structured editors: The MENTOR experience*. DTIC Document. Recuperado a partir de <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA114990>
- Dr. Ernesto Bolaños Rodríguez, A. N. (2012). Muestra y Muestreo.
- Gandecha, K. M., Gondane, V. S., & Shelke, V. R. (s. f.). A Survey on Opinion Mining.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. México, D.F.: McGraw-Hill Education.
- Hewson, C., Vogel, C. M., & Laurent, D. (2016). *Internet research methods* (Second edition). Los Angeles: SAGE.
- INEC. (s. f.). E-commerce Day. Recuperado 24 de octubre de 2016, a partir de <http://www.ecuadorencifras.gob.ec/documentos/web-inec/boletin/E-commerce.pdf>
- Joyce, J. (2016). Bayes' Theorem. En E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016). Metaphysics Research Lab, Stanford University. Recuperado a partir de <https://plato.stanford.edu/archives/win2016/entries/bayes-theorem/>
- Liddy, E. D. (2001). Natural language processing. Recuperado a partir de <http://surface.syr.edu/istpub/63/>
- Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing, 2*, 627–666.
- Martínez Cámara, E., Martín Valdivia, M. T., Perea Ortega, J. M., & Ureña López, L. A. (2011). Técnicas de clasificación de opiniones aplicadas a un corpus en español. Recuperado a partir de <http://rua.ua.es/dspace/handle/10045/18524>
- Mendenhall, W., Beaver, R. J., & Beaver, B. M. (2009). *Introduction to probability and statistics* (13th ed). Belmont, CA: Brooks/Cole, Cengage Learning.

- Naive Bayes text classification. (s. f.). Recuperado 17 de enero de 2017, a partir de <http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html>
- P. de Marchis, G. (2012). La validez externa de las encuestas en la «web» .Amenazas y su control. *Estudios sobre el Mensaje Periodístico*, 18(0). [https://doi.org/10.5209/rev\\_ESMP.2012.v18.40980](https://doi.org/10.5209/rev_ESMP.2012.v18.40980)
- Parzen, E. (1971). *Teoría moderna de probabilidades y sus aplicaciones*. México, D.F. [etc.: Limusa.
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: from theory to algorithms*. New York, NY, USA: Cambridge University Press.
- Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2), 267–307.
- Vázquez, R. (2014, junio 26). La importancia de los sentimientos en redes sociales. Recuperado 23 de octubre de 2016, a partir de <http://www.forbes.com.mx/la-importancia-de-los-sentimientos-en-redes-sociales/>

# **ANEXOS**



# Encuestas de recopilación de expresiones idiomáticas de la jerga ecuatoriana.

Universidad Católica de Santiago de Guayaquil

Facultad de ingeniería

Carrera de Ingeniería en Sistemas Computacionales

Proyecto de Titulación

Autor: Julio Vásconez

Tutor: Ing. Gustavo Molina Flores

Tema: Influencia de las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimientos.

**\*Obligatorio**

## Anexo 2: Cuerpo de encuestas web

Sexo

Hombre

Mujer

¿Qué expresión usaría para expresarse de la mejor manera sobre una persona o institución? \*

Tu respuesta

---

¿Qué expresión usaría para expresar una opinión moderadamente positiva sobre una persona o institución? \*

Tu respuesta

---

¿Qué expresión usaría para expresar una opinión neutral de una persona o institución? \*

Tu respuesta

---

¿Qué expresión usaría para expresar malestar sobre una persona o institución? \*

Tu respuesta

---

¿Qué expresión usaría para expresar la peor opinión sobre una persona o institución? \*

Tu respuesta

---

## Anexo 3: Código Fuente

```
./twitter_api/client.py           Wed Jan 25 13:59:28 2017           1
1: # twitterApi - Twitter Rest API Simplified Client
2: # author: Julio Vasconez
3:
4: import json
5: import requests
6: from operator import itemgetter
7: access_token="AAAAAAAAAAAAAAAAAAAAAH6RxcgAAAAAA%2F%2FHkyD1b8mWSa1JH%2Bpb%2FBzWfKI4%
3DZzGw8dlYzgc4dvdqotvtMbIMeKwql9jLUeNJV3Enyaa4i34Fkb";
8: iteration_limit=100
9: iteration_count=0
10: def getTweets(params):
11:     global iteration_count
12:     print("Iteracion "+ str(iteration_count))
13:     print("Obteniendo Tweets Referentes a "+params["q"]+"...")
14:     headers={"Authorization":"Bearer "+access_token}
15:     result = requests.get("https://api.twitter.com/1.1/search/tweets.json",params=
params,headers=headers)
16:     json_result=json.loads(str(result.text))
17:     return json_result
18:
19:
20: def getAllTweets(params,lowest_id=0,all_statuses=[]):
21:     global iteration_count
22:     global iteration_limit
23:     iteration_count+=1
24:     if(lowest_id!=0):
25:         params["max_id"]=lowest_id-1
26:
27:     new_response=getTweets(params)
28:
29:     if(len(new_response["statuses"]==0) or iteration_count>=iteration_limit:
30:         return new_response["statuses"]
31:     else:
32:         statuses=new_response["statuses"]
33:         lowest_id=min(statuses,key=itemgetter("id"))
34:         return statuses+getAllTweets(params,lowest_id=lowest_id["id"])
35:
36:
```

```
./twitter_api/get_tweets.py       Mon Jan 16 08:12:26 2017           1
1: import json
2: import sys
3: from client import *
4: import unicodedata
5: print("Ingreso Referencia para captura de Tweets")
6: q=input();
7:
8: params={
9:     'q':q,
10:    #'until':'2016-12-28"
11: }
12: result=getAllTweets(params=params)
13:
14: i=0
15: retrieved=[]
16: for t in result:
17:     i+=1
18:     retrieved.append({"i":i,"user":t["user"]["screen_name"],"msg":t["text"],"date"
:t["created_at"]})
19: file_name="tweets-"+q+".json"
20: print("Guardando Archivo: "+file_name+".json...")
21: out_file = open("../res/tweets/"+file_name,"w", encoding='utf-8')
22: json.dump(retrieved,out_file, indent=4)
23: out_file.close()
24: print("Proceso Finalizado")
```

```
1: import nltk
2: import pickle
3: import os
4: from nlp.spanish import stem
5:
6: def extract_feature(w):
7:     return {"word":w}
8:
9: lexicon_lines=open("../res/lexicons/binary_lexicon.txt","r", encoding='utf-8').rea
dlines();
10: classifier_file_name="../res/training/binary_classifier.pickle"
11:
12: labeled_words=[]
13: word_list=[]
14:
15: for line in lexicon_lines:
16:     split=line.replace("\n","").split("\t");
17:     word=stem(split[0].strip().replace(" ","_"))
18:     word_list.append(word);
19:     labeled_words.append((word,split[2]))
20: #construct features
21: feature_set=[(extract_feature(n), sentiment) for (n, sentiment) in labeled_words]
22:
23: def getClassifier():
24:     if not (os.path.isfile(classifier_file_name)):
25:         classifier = nltk.NaiveBayesClassifier.train(feature_set)
26:         f = open(classifier_file_name, 'wb')
27:         pickle.dump(classifier, f)
28:         f.close();
29:     else:
30:         f = open(classifier_file_name, 'rb')
31:         classifier = pickle.load(f)
32:         f.close()
33:     return classifier
34:
35: def getWordList():
36:     return word_list
37:
38: def getFeatureSet():
39:     return feature_set
--
```

```
1: import nltk
2: import pickle
3: import os
4: from nlp.spanish import stem
5:
6: def extract_feature(w):
7:     return {"word":w}
8:
9: lexicon_lines=open("../res/lexicons/binary_lexicon.txt","r", encoding='utf-8').rea
dlines();
10: classifier_file_name="../res/training/multiclass_classifier.pickle"
11:
12: labeled_words=[]
13: word_list=[]
14:
15: #construct features
16: for line in lexicon_lines:
17:     split=line.replace("\n","").split("\t");
18:     word=stem(split[0].strip().replace(" ","_"))
19:     word_list.append(word)
20:     if split[2]=="pos":
21:         sentiment=4
22:     else:
23:         sentiment=2
24:     labeled_words.append((word,sentiment))
25:
26: feature_set=[(extract_feature(n), sentiment) for (n, sentiment) in labeled_words]
27:
28:
29: def getClassifier():
30:     if not (os.path.isfile(classifier_file_name)):
31:         classifier = nltk.NaiveBayesClassifier.train(feature_set)
32:         f = open(classifier_file_name, 'wb')
33:         pickle.dump(classifier, f)
34:         f.close();
35:     else:
36:         f = open(classifier_file_name, 'rb')
37:         classifier = pickle.load(f)
38:         f.close()
39:     return classifier
40:
41: def getWordList():
42:     return word_list
43:
44: def getFeatureSet():
45:     return feature_set
46:
```

```
1: import nltk
2: import pickle
3: import os
4: from nlp.spanish import stem
5:
6: def extract_feature(w):
7:     return {"word":w}
8:
9:
10: lexicon_lines=open("../res/lexicons/binary_lexicon.txt","r", encoding='utf-8').rea
dlines();
11: lexicon_lines_enhance=open("../res/lexicons/ecuador_binary_lexicon.csv","r", encod
ing='latin-1').readlines();
12: classifier_file_name="../res/training/enhanced_binary_classifier.pickle"
13:
14: labeled_words=[]
15: word_list=[]
16:
17: #construct features
18: for line in lexicon_lines:
19:     split=line.replace("\n","").split("\t");
20:     word=stem(split[0].strip().replace(" ","_"))
21:     word_list.append(word);
22:     labeled_words.append((word,split[2]))
23: for line in lexicon_lines_enhance:
24:     split=line.replace("\n","").split(";");
25:     word=stem(split[0].strip().replace(" ","_"))
26:     word_list.append(word);
27:     labeled_words.append((word,split[1]))
28:
29: feature_set=[(extract_feature(n), sentiment) for (n, sentiment) in labeled_words]
30:
31: def getClassifier():
32:     if not (os.path.isfile(classifier_file_name)):
33:         classifier = nltk.NaiveBayesClassifier.train(feature_set)
34:         f = open(classifier_file_name, 'wb')
35:         pickle.dump(classifier, f)
36:         f.close();
37:     else:
38:         f = open(classifier_file_name, 'rb')
39:         classifier = pickle.load(f)
40:         f.close()
41:     return classifier
42:
43: def getFeatureSet():
44:     return feature_set
45: def getWordList():
46:     return word_list
```

```
1: import nltk
2: import pickle
3: import os
4: from nlp.spanish import stem
5:
6: def extract_feature(w):
7:     return {"word":w}
8:
9:
10: lexicon_lines=open("../res/lexicons/binary_lexicon.txt","r", encoding='utf-8').readlines();
11: lexicon_lines_enhance=open("../res/lexicons/ecuador_lexicon.csv","r", encoding='latin-1').readlines();
12:
13: classifier_file_name="../res/training/enhanced_multiclass_classifier.pickle"
14:
15: labeled_words=[]
16: word_list=[]
17:
18: #construct features
19: for line in lexicon_lines:
20:     split=line.replace("\n","").split("\t");
21:     word=stem(split[0].strip().replace(" ","_"))
22:     word_list.append(word)
23:     if split[2]=="pos":
24:         sentiment=4
25:     else:
26:         sentiment=2
27:     labeled_words.append((word,sentiment))
28: for line in lexicon_lines_enhance:
29:     split=line.replace("\n","").split(";");
30:     word=stem(split[0].strip().replace(" ","_"))
31:     word_list.append(word);
32:     labeled_words.append((word,split[1]))
33:
34: feature_set=[(extract_feature(n), sentiment) for (n, sentiment) in labeled_words]
35:
36: def getClassifier():
37:     if not (os.path.isfile(classifier_file_name)):
38:         classifier = nltk.NaiveBayesClassifier.train(feature_set)
39:         f = open(classifier_file_name, 'wb')
40:         pickle.dump(classifier, f)
41:         f.close();
42:     else:
43:         f = open(classifier_file_name, 'rb')
44:         classifier = pickle.load(f)
45:         f.close()
46:     return classifier
47:
48: def getWordList():
49:     return word_list
50:
51: def getFeatureSet():
52:     return feature_set
```

```
1: # -*- coding: utf-8 -*-
2: import classifiers.enhanced_binary_classifier as b_classifier
3: from nlp import spanish as nl
4: classifier = b_classifier.getClassifier()
5: knowed_words=b_classifier.getWordList();
6:
7: while True:
8:     print("Ingrese una expresion para el analisis:")
9:     expression = input()
10:    if expression=="exit":
11:        exit()
12:    tokens=nl.tokenize(expression)
13:    know_count=0;
14:    sentiment_sum=0;
15:    for t in tokens:
16:        if t in knowed_words:
17:            know_count += 1
18:            sentiment=classifier.classify(b_classifier.extract_feature
(t))
19:                print(t+" : "+sentiment)
20:                if sentiment=="pos":
21:                    sentiment_sum += 1
22:                else:
23:                    sentiment_sum -= 1
24:    if know_count>0:
25:        print(know_count)
26:        print (sentiment_sum/know_count*100)
27:    else:
28:        print ("No se puede clasificar")
29:
```



```

1: import tkinter as tk
2: import classifiers.binary_classifier as b_classifier
3: import classifiers.enhanced_binary_classifier as eb_classifier
4: import nlp.spanish as language_tools
5: import matplotlib.pyplot as plot
6: import matplotlib.animation as animation
7: from matplotlib import style
8: import nltk
9: from data.resources import getDataCollection
10: style.use("dark_background")
11: plot.title("Comparacion Algoritmos Binarios")
12: classified_words={"basic":0,"enhanced":0}
13: sentiment_sum={"basic":0,"enhanced":0}
14: classification_features={"basic":[],"enhanced":[]}
15: def classifyTweets(classifier_module,tweets,plt,line_color,name):
16:     classifier=classifier_module.getClassifier()
17:     x=0
18:     y=0
19:     xar=[]
20:     yar=[]
21:     word_list=classifier_module.getWordList()
22:     for t in tweets:
23:         tokens=language_tools.tokenize(t)
24:         sentiment_word_count=0
25:         sum_sentiment=0
26:         for word in tokens:
27:             if word in word_list:
28:                 classified_words[name] += 1
29:                 sentiment_word_count += 1
30:                 sentiment=classifier.classify(classifier_module.extract_feature(word))
31:                 classification_features[name].append((classifier_module.extract_feature(word),sentiment))
32:                 if sentiment=="pos":
33:                     sentiment_sum[name] += 1
34:                     y+=1
35:                 else:
36:                     y-=1
37:                 x += 1
38:                 xar.append(x)
39:                 yar.append(y)
40:         plt.plot(xar,yar,line_color)
41:
42: result=getDataCollection()
43: tweets=[t["msg"] for t in result]
44: #tweets=list(set(tweets))
45: print("*****Inicio de Test*****")
46: print("Cantidad Tweets a Evaluar:"+str(len(tweets))+"\n")
47: classifyTweets(b_classifier,tweets,plot,"g","basic")
48: classifyTweets(eb_classifier,tweets,plot,"b","enhanced")
49: print("Sentimiento segun Clasificador Binario Basico: "+str(sentiment_sum["basic"])/classified_words["basic"])
50: print("Sentimiento segun Clasificador Binario Mejorado: "+str(sentiment_sum["enhanced"])/classified_words["enhanced"])
51: #classifyTweets(emc_classifier,tweets,plot)
52: print("Palabras reconocidas Clasificador Binario Basico: "+str(classified_words["basic"]))
53: print("Palabras reconocidas Clasificador Binario Mejorado: "+str(classified_words["enhanced"]))
54: print("Precision Clasificador Binario Basico contra Mejorado: ",(nltk.classify.accuracy(b_classifier.getClassifier(), classification_features["enhanced"])*100))
55: plot.show()

```

```

1: import classifiers.multiclass_classifier as mc_classifier
2: import classifiers.enhanced_multiclass_classifier as emc_classifier
3: import nlp.spanish as language_tools
4: import matplotlib.pyplot as plot
5: import matplotlib.animation as animation
6: from matplotlib import style
7: import nltk
8: from data.resources import getDataCollection
9:
10: style.use("dark_background")
11: plot.title("Comparacion Algoritmos Multiclase")
12: classified_words={"basic":0,"enhanced":0}
13: sentiment_sum={"basic":0,"enhanced":0}
14: classification_features={"basic":[],"enhanced":[]}
15: def classifyTweets(classifier_module,tweets,plt,line_color,name):
16:     classifier=classifier_module.getClassifier()
17:     x=0
18:     y=0
19:     xar=[]
20:     yar=[]
21:     word_list=classifier_module.getWordList()
22:     for t in tweets:
23:         tokens=language_tools.tokenize(t)
24:         for word in tokens:
25:             if word in word_list:
26:                 classified_words[name] += 1
27:                 sent=int(classifier.classify(classifier_module.extract_
ract_feature(word)))
28:                 classification_features[name].append((classifier_m
odule.extract_feature(word),sent))
29:                 y += sent-3
30:                 sentiment_sum[name] += sent
31:
32:
33:                 x += 1
34:                 xar.append(x)
35:                 yar.append(y)
36:                 plt.plot(xar,yar,line_color)
37:
38:
39:
40:
41: result=getDataCollection()
42: tweets=[t["msg"] for t in result]
43: #tweets=list(set(tweets))
44: print("*****Inicio de Test*****")
45: print("Cantidad Tweets a Evaluar:"+str(len(tweets)))
46: classifyTweets(mc_classifier,tweets,plot,"g","basic")
47: classifyTweets(emc_classifier,tweets,plot,"b","enhanced")
48: #classifyTweets(emc_classifier,tweets,plot)
49: print("Sentimiento segun Clasificador Multiclase Basico: "+str(sentiment_sum["bas
ic"]/classified_words["basic"]))
50: print("Sentimiento segun Clasificador Multiclase Mejorado: "+str(sentiment_sum["e
nhanced"]/classified_words["enhanced"]))
51: print("Palabras reconocidas Clasificador Multiclase Basico: "+str(classified_words
["basic"]))
52: print("Palabras reconocidas Clasificador Multiclase Mejorado: "+str(classified_wor
ds["enhanced"]))
53: print("Precision Clasificador Multiclase Basico contra Mejorado: ",(nltk.classify.
accuracy(mc_classifier.getClassifier(),classification_features["enhanced"]))*100)
54: plot.show()

```

./nlp/spanish.py

Wed Mar 15 13:49:39 2017

1

```
1: # -*- coding: utf-8 -*-
2: import nltk
3: from nltk.corpus import stopwords
4: from nltk import word_tokenize
5: from nltk.stem import SnowballStemmer
6: from string import punctuation
7:
8: spanish_stopwords = stopwords.words('spanish')
9: stemmer = SnowballStemmer('spanish')
10:
11: non_words = list(punctuation)
12: non_words.extend(['Ã', 'Ãi'])
13: non_words.extend(map(str, range(10)))
14:
15: def stem(word):
16:     return stemmer.stem(word)
17:
18: def stem_tokens(tokens):
19:     stemmed = []
20:     for item in tokens:
21:         stemmed.append(stem(item))
22:     return stemmed
23:
24: def tokenize(text):
25:     text = ''.join([c for c in text if c not in non_words])
26:     tokens = word_tokenize(text)
27:     tokens = stem_tokens(tokens)
28:     bigrams=[ x.lower()+"_"+y.lower() for (x,y) in zip(tokens,tokens[1:])]
29:     trigrams=[ x.lower()+"_"+y.lower() for (x,y) in zip(bigrams,tokens[2:])]
30:     tokens=[t.lower() for t in tokens ]+bigrams+trigrams
31:     return tokens
32:
```

./data/resources.py

Tue Jan 31 05:51:32 2017

1

```
1: import os
2: import json
3: def getDataCollection():
4:     path="./res/tweets/"
5:     dirs=os.listdir(path)
6:     print ("Escoja la coleccion de tweets para el proceso:")
7:     menu_index=1;
8:     for file in dirs:
9:         print (str(menu_index)+" "+file)
10:        menu_index+=1
11:    selected=input()
12:    if "json" in dirs[int(selected)-1]:
13:        json_data=json.loads(open(path+"/"+dirs[int(selected)-1]).read())
14:    else:
15:        json_data=[{"msg":m} for m in open(path+"/"+dirs[int(selected)-1])
.readlines() ]
16:    return json_data
```



## DECLARACIÓN Y AUTORIZACIÓN

Yo, **Vásconez Yulán, Julio Oswaldo**, con C.C: # **0923526545** autor del trabajo de titulación: **Influencia de las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimientos** previo a la obtención del título de **Ingeniero en Sistemas Computacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 21 de Marzo de 2017

f. \_\_\_\_\_

Nombre: **Vásconez Yulán, Julio Oswaldo**

C.C: **0923526545**



<b>REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA</b>			
<b>FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN</b>			
<b>TÍTULO Y SUBTÍTULO:</b>	Influencia de las expresiones idiomáticas propias de una jerga sobre algoritmos de Análisis de Sentimientos.		
<b>AUTOR</b>	Julio Oswaldo, Vásconez Yulán		
<b>TUTOR</b>	Ing. Gustavo, Molina Flores, Mgs		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Ingeniería		
<b>CARRERA:</b>	Carrera de Ingeniería en Sistemas Computacionales		
<b>TITULO OBTENIDO:</b>	Ingeniero en Sistemas Computacionales		
<b>FECHA DE PUBLICACIÓN:</b>		<b>No. DE PÁGINAS:</b>	69
<b>ÁREAS TEMÁTICAS:</b>	Hardware, Software, Redes y Comunicaciones		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Sentimiento, Opinión, Análisis, Subjetividad, Aprendizaje-Automático, Léxico		
<b>RESUMEN/ABSTRACT</b> (150-250 palabras): Considerando lo importante que puede llegar a ser para una persona, institución o producto, en la actualidad; conocer el nivel de aceptación o el sentimiento que genera en las en la sociedad digital, surge el Análisis de Sentimiento como respuesta, con la finalidad de procesar una gran cantidad de opiniones de manera automática, haciendo uso de técnicas de clasificación. El desarrollo de un prototipo de Análisis de Sentimiento basado en léxicos permite enfrentar la problemática mencionada al añadir expresiones idiomáticas propias de la jerga ecuatoriana a las consideraciones del algoritmo de clasificación, mismas que fueron recopiladas a través de una investigación cualitativa dentro de una muestra de usuarios digitales de Ecuador.			
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR:</b>	<b>Teléfono:</b> +593-4-2214845	<b>E-mail:</b> jvasconez28@gmail.com	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::</b>	<b>Nombre: Valencia Macias, Lorgia del Pilar</b>		
	<b>Teléfono: +593-4-2206950 ext 1020</b>		
	<b>E-mail: lorgia.valencia@cu.ucsg.edu.ec</b>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			