
UNIVERSIDAD CATÓLICA de Colombia

# APLICACIÓN DEL APRENDIZAJE AUTOMÁTICO EN LA CLASIFICACIÓN DE TEXTOS CORTOS: UN CASO DE ESTUDIO EN EL CONFLICTO ARMADO COLOMBIANO.

Presentado por:
**Juan Pablo Páramo Lozada.**
**Cesar Augusto Espitia Betancourt.**

Universidad Católica de Colombia
Facultad de ingeniera
Departamento de ingeniería de sistemas
Bogotá Colombia
2018

UNIVERSIDAD CATÓLICA de Colombia

# APLICACIÓN DEL APRENDIZAJE AUTOMÁTICO EN LA CLASIFICACIÓN DE TEXTOS CORTOS: UN CASO DE ESTUDIO EN EL CONFLICTO ARMADO COLOMBIANO.


Presentado por:

**Juan Pablo Páramo Lozada. Código: 625381**

**Cesar Augusto Espitia Betancourt. Código: 625373**

Asesores:

Raúl Ernesto Menéndez Mora, PhD

Erika Paola Holguín Ontiveros.

Universidad Católica de Colombia

Facultad de ingeniera

Departamento de ingeniería de sistemas

Bogotá Colombia


2018

Nota de aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos Exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de Ingenieros de Sistemas.

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado 2

Bogotá (01, 11, 2018)

## TABLA DE CONTENIDO

## LISTA DE ILUSTRACIONES

## LISTA DE TABLAS

## LISTA DE ABREVIATURAS

**CRF:** (Conditional random field) o Campo aleatorio condicional.
**ELN:** Ejército de Liberación Nacional.
**FARC:** Fuerza Alternativa Revolucionaria del Común.
**HI:** Historia de usuario.
**FN:** Falso negativo.
**FP:** Falso positivo.
**TP:** Verdadero positivo.
**TN:** Verdadero negativo.
**TXT:** Archivo de texto.
**URL:** Localizador uniforme de recursos.

## RESUMEN

El presente proyecto busca abordar la problemática del conflicto armado que posee Colombia, lleva alrededor de 50 años en guerra con diferentes grupos armados, uno de ellos y con el cual el conflicto presento repercusiones negativas para el país fue las FARC-EP (Fuerzas Armadas Revolucionarias de Colombia-Ejército del Pueblo) que, aunque los acuerdos de paz silenciaron las balas no silencio a sus representantes y las disputas continúan, pero ahora son en las redes sociales, permitiéndoles expresar su manera de pensar o difundir aspectos políticos que tienen que ver con el proceso de paz, por lo que es importante saber la orientación de un tweet para no entorpecer el proceso que se lleva con el ELN (Ejército de Liberación Nacional).

Para abordar esta problemática existen diferentes técnicas que permiten analizar la posición en que se encuentra cada actor del conflicto basándose en sus publicaciones (Tweets), el aprendizaje automático es una rama de la inteligencia artificial y tiene como característica que permite el procesamiento del lenguaje natural, es decir, que la maquina pueda interactuar de manera eficiente con el lenguaje humano, también se encuentra el pre-procesamiento de textos, este proceso permite que la maquina entienda con mayor facilidad y eficiencia el lenguaje humano, esto es importante debido a que las publicaciones de Twitter se realizan de manera informal por lo que el autor de cada publicación puede incluir elementos no propios del idioma, esto causa que el algoritmo pierda eficiencia en el aprendizaje, luego de esto mediante técnicas como “Bolsa de palabras” entre otras, este texto se puede convertir en elementos numéricos para que puedan ser procesados por un algoritmo de aprendizaje automático.

Los algoritmos de aprendizaje se dividen en 2 categorías principales, el aprendizaje supervisado y el aprendizaje no supervisado, para saber cuál categoría seguir se debe tomar en cuenta la estructura de los datos obtenidos, si los datos ya están clasificados se debe seguir el aprendizaje supervisado, si no cuentan con una clasificación, se debe seguir el aprendizaje no supervisado, en cada categoría existen algoritmos específicos con los cuales se puede realizar el entrenamiento de la máquina, también existen herramientas que cuentan con estos algoritmos implementados que facilitan la tarea de la clasificación y son integrables a lenguajes de programación por lo que es viable desarrollar una interfaz gráfica que permita al usuario interactuar con los algoritmos y permita sacar conclusiones con respecto a varios algoritmos combinados con diferentes técnicas.



**Palabras clave:** Conflicto armado, Red social, Algoritmos, Aprendizaje automático, lenguaje natural, pre-procesamiento de textos, procesamiento de textos.

## ABSTRACT

This project seeks to address the problem of armed conflict that Colombia has, there has been around 50 years in war with different armed groups, one of them and with which the conflict presented negative repercussions for the country was the FARC-EP (Revolutionary Armed Forces of Colombia-People's Army) that, although the peace agreements silenced the bullets not silence their representatives and the disputes continue, but now they are in social networks, allowing them to express their way of thinking or disseminate political aspects that have to do with the peace process, so it is important to know the orientation of a tweet so as not to obstruct the process that is carried out with the ELN (National Liberation Army).

To address this problem there are different techniques to analyze the position in which each actor of the conflict is based on their publications (Tweets), machine learning is a branch of artificial intelligence and has the characteristic that allows the processing of natural language, that is, where machine can interact efficiently with human language, also is the pre-processing of texts, this process allows the machine to understand human language more easily and efficiently, this is important because the publications of Twitter carry out informally language so that the author of each publication can include elements that are not specific to the language, this causes the algorithm to lose efficiency in learning, after this through techniques such as "Word Bag" among others, this text it can be converted into numerical elements so that they can be processed by an algorithm of automatic learning

The learning algorithms are divided into 2 main categories, supervised learning and unsupervised learning, to know which category to follow the structure of the data obtained must be taken into account, if the data is already classified supervised learning must be followed, if they do not have a classification, the unsupervised learning should be followed, in each category there are specific algorithms with which the machine training can be carried out, there are also tools that have these algorithms implemented that facilitate the task of classification and they are integrable to programming languages, so it is feasible to develop a graphical interface that allows the user to interact with the algorithms and to draw conclusions regarding several algorithms combined with different techniques.



**Keywords:** Armed conflict, Social network, Algorithms, Machine learning, natural language, pre-processing of texts, word processing.

# INTRODUCCIÓN

## 1. GENERALIDADES

El presente proyecto de grado aborda la problemática del pre procesamiento de textos cortos extraídos de la red social Twitter para el posterior análisis y clasificación de la orientación pacifista, guerrerista o neutra de dicho texto. Los textos extraídos pertenecen a comunicados emitidos en la red social por los principales actores del conflicto colombiano.

Este proyecto forma parte de un proyecto institucional desarrollado entre la facultad de psicología y la de ingeniería, titulado: *"Efectividad de un protocolo de re-experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política*". Este trabajo tiene sus cimientos en otro proyecto del programa de Ingeniería de Sistemas el cual se centró en la extracción de los textos cortos de la red social Twitter y su almacenamiento en archivos de texto plano (documentos .txt). Los datos extraídos por el proyecto antes mencionado, constituyen las entradas de este trabajo que se está presentando.

El objetivo principal del proyecto es clasificar la orientación de un texto corto de manera eficaz, utilizando diferentes algoritmos de aprendizaje automático. Sin embargo, el estudio de diversas técnicas de pre-procesamiento de textos y su impacto en el rendimiento de los algoritmos de clasificación constituye una parte no menos importante en este trabajo.

La metodología del proyecto consiste en obtener la base de datos la cual contiene toda la información de cada tweet como lo son nombre de usuario, identificador (id), el texto, fecha, entre otros diferentes atributos. El campo texto de cada uno de estos tweets ha sido evaluado por 4 jurados, los cuales son psicólogos de la Universidad Católica de Colombia. Estos psicólogos definieron la orientación de cada texto.

Una vez procesados los datos, se utilizan los algoritmos de aprendizaje automático más utilizados para predecir la orientación del texto. Se comparan los resultados obtenidos con los resultados ya predefinidos por los expertos y así poder calcular la eficiencia de los algoritmos usados.

## 1.1. ANTECEDENTES

El presente proyecto precede tres proyectos los cuales son:

- Desarrollo y aplicación de una herramienta de extracción y almacenamiento de datos de Twitter a un contexto social de violencia política.
- Efectividad de un protocolo de re-experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política.
- Desarrollo de un componente de analítica para la clasificación de textos cortos dirigido a un proyecto institucional e integrable en una plataforma web.

Los dos primeros proyectos trabajan en conjunto. El primer proyecto corresponde a la facultad de ingeniería de sistemas el cual se basa en el desarrollo de una herramienta de extracción de publicaciones de la red social twitter, el segundo proyecto mencionado fue realizado en la facultad de psicología, este proyecto estudia a profundidad el tema del conflicto armado, también estudia las publicaciones realizadas en la red social twitter emitidas por los actores del conflicto armado, este análisis permite clasificar los tweets en 3 categorías (Pacifista, Guerrerista, Neutra). Se aclara que las publicaciones fueron escogidas por la facultad de psicología para su posterior extracción con la herramienta elaborada por la facultad de sistemas y seguidamente la facultad de psicología procede con la clasificación de textos cortos en su respectiva categoría. El tercer proyecto desarrolla un componente en Python que permita clasificar estos tweets en categorías.

## 1.2. PLANTEAMIENTO DEL PROBLEMA

### 1.2.1. Descripción del problema.

Colombia lleva alrededor de 50 años en guerra con diferentes grupos armados, principalmente con las FARC-EP (Fuerzas Armadas Revolucionarias de Colombia-Ejército del Pueblo) que llevan exactamente 52 años de conflicto, inicio el 14 de mayo de 1964 y finalizo con la firma de los acuerdos de paz el 24 de noviembre de 2016. Se logró en la Habana Cuba, un acuerdo en el cual se acordaba el cese al fuego bilateral entre otros, los cuales tienen como objetivo la paz en Colombia. El 28 de agosto de 2017 se instauró a las FARC como un partido político, conservando su acrónimo, pero con diferente significado, partido político Fuerza Alternativa Revolucionaria del Común (FARC) (Radio, 2017). Los acuerdos que se lograron con el tratado de paz, se están intentando replicar con otros grupos revolucionaron, principalmente con el ELN (Ejército de Liberación Nacional).

Las secuelas del problema aún continúan y en la actualidad varios actores del conflicto utilizan las redes sociales para expresar su manera de pensar o difundir aspectos políticos que tienen que ver con el proceso de paz y con la sociedad en general. Estos actores utilizan medios como la red social Twitter para generar en los lectores, varios estados de opinión. Básicamente, estos actores del conflicto con sus comunicados de texto corto generan estados de opinión no necesariamente positivos para la instauración de una cultura de paz.

Con estos comunicados, los actores del conflicto buscan plasmar las ideologías que siguen, los grupos que generan violencia están enfocados en minimizar el impacto emocional negativo que han generado sus acciones y el gobierno colombiano como mediador.

Las diferentes ideologías que siguen los actores del conflicto generan un choque entre las diferentes partes, lo que resulta que cada uno de ellos ataque al otro. Esto es realizado por diferentes medios, pero para el caso de estudio propuesto se va a observar los comunicados de textos cortos. El objetivo de estos comunicados es legitimar o deslegitimar los diferentes participantes buscando que actores como lo es la sociedad colombiana, cambien su concepto frente a otro actor del conflicto.

Se analizan los comunicados de las Fuerzas Armadas Revolucionarias de Colombia (FARC-EP) y el Gobierno Colombiano, en estos dos grupos hay líderes que generaran con sus publicaciones pensamientos guerreristas y no necesariamente son las FARC-EP. Las diferentes intenciones de cada actor es importante analizarlas para que se pueda tomar la decisión más acertada y no solo lograr un proceso de paz, si no que se pueda, mediante el análisis de comunicados de textos cortos, solucionar diferentes problemas sociales a partir de la clasificación de la orientación de cada uno de ellos.

Este problema es reconocido por la facultad de psicología y en el caso de estudio desarrollado identifico que los comunicados realizados principalmente en la red social twitter tienen un impacto social en los lectores, por esto clasificar los tweets es importante para saber si alguno en particular tiene una orientación pacifista guerrerista o neutra y esto se puede lograr utilizando Machine Learning o en español aprendizaje automático.

El aprendizaje automático es una rama de la inteligencia artificial la cual se aplica en el desarrollo del proyecto por la necesidad de que una máquina sea la que realice la clasificación y no siempre la lleve a cabo un jurado de psicólogos. Con los tweets ya clasificados, se puede entrenar una máquina para que sea esta la que realice la predicción de la orientación de un comunicado corto.

### 1.2.2. Pregunta científica orientada a la formulación del problema.

¿Cómo se puede aplicar el aprendizaje automático para identificar la orientación pacifista, guerrerista o neutra de un comunicado corto emitido por un actor del conflicto armado colombiano?

## 1.3. OBJETIVOS

### 1.3.1. Objetivo general.

Clasificar textos cortos (tweets), extraídos de la red social twitter, en una orientación guerrerista, pacifista o neutra para un caso de estudio en el conflicto armado colombiano, teniendo en cuenta la aplicación del aprendizaje automático (machine learning).

### 1.3.2. Objetivos específicos.

1. Realizar una revisión sistemática de la literatura para estructurar un estado de la cuestión relacionado con:
    a) Algoritmos de clasificación de textos cortos.
    b) Técnicas de procesamiento de lenguaje natural más usadas para el pre-procesamiento de textos cortos.
2. Identificar los entornos o plataformas, relacionados con la implementación de procesos de clasificación de textos cortos.
3. Identificar los algoritmos más usados para la clasificación de textos cortos, teniendo en cuenta los entornos o plataformas verificados para la implementación de los mismos.
4. Comparar la eficiencia de los algoritmos seleccionados, a partir del caso de estudio.

## 1.4. JUSTIFICACIÓN

El presente proyecto puede ser utilizado como base para las negociaciones que actualmente se tienen con el ELN debido a que se analiza la orientación de los comunicados emitidos por actores del conflicto armado permitiendo saber si el proceso va por buen camino al igual que tomar acciones correctoras para un caso contrario, la facultad de psicología realizó un análisis que le permitió elegir tweets relevantes de actores del conflicto y clasificarlos en tres categorías principales (pacifista, guerrerista, neutral).

Al utilizar el Machine Learning se realiza el entrenamiento de una máquina que realizará la clasificación de textos cortos la cual es útil porque permite analizar y poder determinar si el comunicado realizado por actores del conflicto, extraído de la red social Twitter, tiene una orientación pacifista, guerrerista o neutra. Los estudios psicológicos y de tendencias que se generen de esta clasificación automática pudieran impactar positivamente a otras negociaciones de conflictos aún en procesos como son las negociaciones con el ELN.

Al comparar la eficiencia de diferentes algoritmos de aprendizaje, se podrá escoger el más adecuado para implementar el proceso de identificación de la orientación de un comunicado de texto corto y poder así implementarlo en una plataforma, paquete, entorno o arquitectura. Una vez escogido e implementado el algoritmo y la plataforma más adecuada, se procesan diferentes comunicados de los cuales, se podrá saber la orientación de dicho texto y sabiendo la orientación que instauro el grupo de ínter-jueces conformado por psicólogos de la Universidad Católica de Colombia.

## 1.5. DELIMITACIÓN

### 1.5.1. Espacio.

La localización del presente proyecto será en Bogotá D.C específicamente en la universidad católica de Colombia.

### 1.5.2. Tiempo.

El proyecto se tiene planeado ser desarrollado en 6 meses, en la sección de planeación del proyecto se detalla todo el proceso a seguir durante las etapas y los tiempos de las mismas.

### 1.5.3. Contenido.

El contenido del proyecto se enfoca en el aprendizaje de maquina en el cual se evalúa las técnicas y algoritmos acordes a la problemática que se plantea, adicional a esto las técnicas deberán ser recientes.

### 1.5.4. Alcance.

Utilizar los algoritmos de clasificación más utilizados considerando lo reportado en la revisión del estado de la cuestión. Tener en cuenta que el conjunto de entrenamiento consiste en una base de datos que suministro la facultad de psicología donde se tienen los acuerdos ínter-jueces. La eficiencia de la clasificación está sujeta a la cantidad de ejemplos disponibles.

### 1.5.5. Impacto y/o viabilidad.

El proyecto tiene características sociales y busca ayudar en temas sociales, como lo es en la paz de la República de Colombia, ayudando de manera directa en los tratados de paz, especialmente con los futuros acuerdos de paz que se están negociando con la organización guerrillera ELN, al informarle al ente competente y/o que utilice el componente en determinar si un comunicado corto tiene una orientación pacifista, guerrerista o neutra y así saber si la posición en la que se encuentra o el pensamiento que tiene el emisor del tweet.

## 1.6. MARCO REFERENCIAL

### 1.6.1. Marco conceptual.

1.6.1.1. Algoritmos de clasificación de textos.

Están diseñados con la finalidad de automatizar procedimientos como la tipologización textual que es realizada por humanos, la cual, en base a diferentes criterios, clasifica los textos (Venegas, 2007), por lo que basados en estos criterios se busca el algoritmo que más se ajuste y se automatiza la tarea.

1.6.1.2. Bigramas.

Son estructuras que permiten almacenar duplas de datos, utilizados para no perder la semántica de una oración (Le & Mikolov, 2014).

1.6.1.3. F-measure.

Es una medida armónica de precisión (P) y recall (R), la cual está definida por la siguiente ecuación, donde P es la precisión, que es la fracción de instancias importantes entre las instancias recuperadas y R es la recuperación, que es la fracción de instancias importantes que se han recuperado sobre el total de estas (Sasaki & Fellow, 2007))

$$F = 2 * \frac{P * R}{P + R}$$

1.6.1.4. FN.

O falso negativo, son aquellas instancias o predicciones que son correctas o positivas y el sistema las reconoce como falsas o negativas. (Bouckaert et al., 2016).

1.6.1.5. FP.

O falso positivo, son aquellas instancias que son negativas, pero el sistema afirma que no lo son. (Bouckaert et al., 2016).

1.6.1.6. IDF.

Es la frecuencia inversa del término, es decir, a medida que aparece el término disminuye su peso.

#### 1.6.1.7. Matriz de confusión.

En una matriz de confusión se puede visualizar los errores obtenidos al emplear un clasificador, como su nombre lo indica es una matriz la cual se puede representar con una tabla, en la cual se puede analizar si al emplear el clasificador este está confundiendo las clases que interactúan. (Corso & Lorena, 2010)

#### 1.6.1.8. MCC.

Se utiliza como una medida de la calidad de las clasificaciones binarias. Se tiene en cuenta los valores TP, FP y FN, es considera como una medida equilibrada y se puede usar, aunque las clases sean de tamaños diferentes.

#### 1.6.1.9. Mindfulness.

Es un término utilizado en la psicología el cual no tiene traducción al idioma español, consiste en pretender que la persona se centre en el presente, en el momento actual, prestando total atención y teniendo conciencia plena, en vez de vivir en un mundo irreal soñando despierto (Miguel Ángel Vallejo Pareja, 2006)

#### 1.6.1.10. Pre-procesamiento de textos.

Consiste en mediante algún tipo de técnicas transformar los textos teniendo como resultado una representación estructurada que facilite el análisis de los mismos (Centro Nacional de Información de Ciencias Médicas. & Cabrera-Gato, 2007), para lograr esta estructuración se elimina cierta información textual que no es relevante para la finalidad del proyecto (Gálvez, 2008).

#### 1.6.1.11. PRC Área.

PCR (gráfico de precisión de recuperación), muestra los valores de precisión para los valores de sensibilidad (recuperación), esta grafica evalúa todo el modelo, principalmente informa cómo se comporta el clasificador en una clase

#### 1.6.1.12. Precision.

Mide la cantidad de términos (instancias) reconocidos correctamente frente al total de términos establecidos clasificadas como una clase, busca responder la siguiente pregunta ¿Qué fracción de los positivos predichos son realmente positivos? (Corso & Lorena, 2010) y la manera de responder la pregunta es utilizando la siguiente ecuación:

$$Precision = \frac{tp}{tp + fp}$$

1.6.1.13. Recall.

Es la probabilidad de identificar correctamente los términos respecto a un total de términos reales, se utiliza para validar. si la variable de sensibilidad o TP realmente se predijo correctamente (Bouckaert et al., 2016), para esta validación se sigue la siguiente ecuación:

$$Recall = \frac{tp}{tp + fn}$$

1.6.1.14. ROC Área.

El ROC (curvas de recuperación de precisión), es un gráfico que permite analizar cuál es el rendimiento promedio del clasificador evaluado y este proceso es realizado dividiendo los datos en clases como: TP, TN, FP y FN.

1.6.1.15. TF.

Es la frecuencia de término, es decir, cuantas veces aparece determinado termino en un documento.

1.6.1.16. TF-IDF.

Es el resultado de la multiplicación de TF e IDF. ((Adam Marcus y Eugene Wu, 2012))

1.6.1.17. TN.

O verdadero negativo, son aquellas instancias que son negativas y el sistema afirma que lo son. (Bouckaert et al., 2016)

1.6.1.18. TP.

O verdadero positivo, son aquellas instancias que son positivas y el sistema afirma que lo son, esta variable también es conocida como sensibilidad. (Bouckaert et al., 2016)

1.6.1.19. Tweet (Tuit).

Son todas aquellas publicaciones que pueden enviar los usuarios registrados en Twitter, los tweets son usados para que los usuarios de la red social se puedan expresar y tienen un máximo de 280 caracteres. (Diccionario Cambridge, 2018)

### 1.6.2. Marco teórico.

El mundo ha visto cambios significativos en el ámbito de la tecnología en los últimos años, la disciplina “Inteligencia Artificial” se ha venido estudiando desde alrededor de 1936 con Alan Turing consolidándose en 1956 como una rama independiente en la informática. Esta ha sido una de las ramas que más estudios y avances ha tenido en los últimos años y, además, se ha identificado la gran variedad de aplicaciones que tienen estos estudios en el diario vivir. Algunos ejemplos de estas aplicaciones son: la robótica, educación, ingeniería, procesamiento y análisis de datos, finanzas, desarrollo de software en general, sistemas inteligentes, entre otras muchas aplicaciones.

Hoy en día se habla bastante del termino big data no solo en ámbitos de educación si no que ya se ha expandido a el ámbito empresarial e industrial, este término ha surgido no hace muchos años pero su aplicación ha avanzado en grandes cantidades de manera muy rápida por lo que se considera que su avance ha sido descontrolado, también al tener un avance tan descontrolado no se ha consolidado una definición especifica global, cada autor define a su consideración y a pesar de que el tema es relacionado no es un estándar, (De Mauro, Greco, & Grimaldi, 2015) realizaron una investigación de diferentes autores y tomaron varias definiciones para tratar de sacar una conclusión de la definición global de big data y en su trabajo concluyeron que big data es “la representación de los activos de información caracterizados por tal alto volumen, velocidad y variedad que requieren tecnología y métodos analíticos específicos para su transformación en valor.”, esta información puede ser tratada de muchas maneras y existen muchas técnicas y metodologías que permiten tratar los datos para un fin, una de estas es el Machine learning.

Machine learning, o aprendizaje automático, consiste en el desarrollo de diferentes técnicas para la identificación de patrones entre numerosos datos relacionados, adaptarse a cambios y ayudar a la mejora del rendimiento (Arcila-Calderón, Barbosa-Caro, & Cabezuelo-Lorenzo, 2016). La finalidad es enseñarle a una maquina a realizar una tarea específica mediante la utilización de datos históricos, estos datos le permiten a la maquina realizar una tarea de aprendizaje de manera similar como los humanos la hacen.

El aprendizaje de maquina cuenta con modelos los cuales permiten realizar una tarea dada, los modelos más conocidos son:

Modelos geométricos, estos modelos toman como base datos numéricos y estos datos los ubica en un espacio de instancias, este espacio de instancias es mejor conocido como planos cartesianos, estos pueden ser bidimensionales o multidimensionales dependiendo de la tarea a realizar, para saber qué espacio se utiliza se debe analizar el conjunto de datos de entrenamiento, dependiendo de las

variables con las que cuenta el problema se tendrá dicha cantidad de dimensiones, luego de esto el modelo identifica mediante la utilización de la geometría como líneas, planos, híper-planos y distancias, como está dividido el conjunto de datos permitiendo realizar un análisis de una manera sencilla (“Los Modelos Geométricos (Modelos parte I)",2017).

Modelos probabilísticos, estos tienen un enfoque estadístico por lo que utiliza varias de las funciones de probabilidad, en un comienzo esta función de probabilidad es desconocida por lo que al analizar los datos se empieza a conocer acerca de la forma de la distribución, al encontrar esta distribución permite resolver el supuesto de hallar la probabilidad de una variable que no se conoce (X) con respecto a las variables que se conocen (Y), es decir, nuestro set de datos, al conocer la distribución se podrán realizar predicciones, un clásico ejemplo de esto son los correos spam, allí se tiene un conjunto de correos spam considerados (X) y llega un correo (Y), si este cumple con las características del conjunto (Y) es etiquetado como spam, si no, (Y) es clasificado como un correo común (“Modelos Probabilísticos (Modelos parte 2)", 2017).

Modelos logísticos, la naturaleza de este modelo es de tipo algorítmica, este modelo cuenta con una característica importante y es que se pueden convertir, mediante ciertas reglas, en un lenguaje comprensible por el humano, estos modelos son representados en forma de árbol, estos árboles están compuestos por nodos y estos nodos a su vez pueden tener sub-nodos, estos nodos están etiquetados, los últimos nodos del árbol son la salida del modelo, allí contienen las clases, probabilidades, valores, a las que se necesite clasificar ciertos datos, es importante decir que la base de estos modelos es dividir el problema en problemas más pequeños por lo que estos provienen del algoritmo “Divide y vencerás”, una característica importante de estos modelos es que se debe elegir bien la raíz del árbol para que en un futuro el árbol no posea nodos que no va a recorrer, es decir, nodos que no sean funcionales, estos nodos no funcionales hacen que el modelo sea más lento y menos eficiente, diferente a los modelos geométricos y probabilísticos, este modelo es de fácil inspección por humanos (“Modelos Logicos (Modelos parte 3)", 2017).

Modelos de agrupamiento, estos modelos de agrupamiento busca dividir un set de datos ubicado en cualquiera de los 3 modelos anteriores en K grupos de datos, estos tienen la característica que cada uno de los K grupos debe está formado por los datos más similares, para que el modelo pueda identificar cuáles son estos datos utiliza funciones de similitud o funciones de distancia, para este modelo es importante encontrar “ruido” en el set de datos para poder lograr una eficiencia mayor y lograr resultados esperados, en textos los caracteres especiales pueden ser considerados ruido por lo que se deben remover y luego implementar este modelo, con esto se asegura que el modelo detecte correctamente los grupos y estime el numero correcto de los mismos (Pascual, Pla, & Sánchez, 2007) estos se pueden dividir en paramétricos y no paramétricos, los paramétricos construyen

una hipótesis que dice que los datos de entrada cuentan con una distribución de probabilidad que puede ser descrita mediante parámetros, computacionalmente estos son más eficientes que los no paramétricos pero al basarse en una hipótesis hace que no sea tan robusto como los no paramétricos los cuales se basan en que a medida que el conjunto de datos crece, la complejidad del espacio crece y no puede considerar algunos parámetros, considera todo (“Repaso didáctico sobre machine learning",2015).

El machine learning cuenta con dos líneas de algoritmos principales las cuales son el aprendizaje supervisado y el aprendizaje no supervisado: El aprendizaje no supervisado no tiene un conocimiento previo acerca de los resultados que se van a obtener después de aplicar el algoritmo. Se proporciona el conjunto de datos de entrada para tener como resultado el descubrimiento de patrones, en los datos, sus características, categorías, entre otros diferentes atributos (Arcila-Calderón et al., 2016), y se encuentra la metodología de aprendizaje supervisado tiene un conjunto de datos de entrada previamente evaluados los cuales van servir como un conjunto de entrenamiento (Pérez Abelleira & Cardoso, 2010) y el objetivo va a ser asignar una clasificación a un segundo grupo a partir del primero, el aprendizaje supervisado va a ser la metodología que se va a seguir en el proyecto ya que, como se mencionó anteriormente el proyecto consiste en que a partir de una base de datos la cual contiene cierta cantidad de tweets, este conjunto de entrenamiento va a permitir enseñarle al algoritmo a realizar predicciones por lo que esta técnica se ajusta con los objetivos del proyecto.

### 1.6.3. Marco geográfico.

El proyecto se desarrolla en la sede El Claustro de la universidad católica de Colombia localizada específicamente en la Diagonal 46 A # 15 B – 10 de la ciudad de Bogotá.

Ilustración 1. Ubicación geográfica.


Fuente: Google Maps - https://maps.google.com/

### 1.6.4. Marco demográfico.

El proyecto está dirigido a la población involucrada en el proceso de paz y reconciliación de Colombia, va dirigido tanto a los actores activos como lo son los ex-miembros de las FARC, ELN y el gobierno colombiano, como a los actores pasivos por ejemplo los familiares de las víctimas.

## 1.7. ESTADO DEL ARTE

El pre-procesamiento de textos es una parte fundamental para poder clasificar eficientemente el sentimiento del mismo, este proceso debe ser más riguroso cuando el texto es considerado corto ya que se disponen de menos elementos (palabras) que permitan determinar la polaridad, además existe el factor de que en la red social twitter se utilizan URLs, lenguaje informal, errores humanos, emojis, entre otros factores que son irrelevantes y hacen complejo el trabajo de clasificación para los algoritmos. La finalidad del pre-procesamiento es minimizar todos estos factores para aumentar la eficiencia del algoritmo al clasificar. Algunas técnicas que se utilizan actualmente son:

### 1.7.1. Tokenizadores.

(KDnuggets, 2017) define este proceso, también conocido como segmentación de texto, como una técnica la cual se basa en romper cadenas de texto largas en cadenas más cortas y estas cadenas cortas en palabras las cuales son llamadas tokens, (Paute, Soroa, & López, 2016) dice que para hallar estos tokens se debe analizar el problema y decidir la forma más eficiente de dividir estas frases, usualmente para esto se utilizan los signos de puntuación que son llamados delimitadores, después, los delimitadores se reemplazan por un espacio en blanco para lograr que las palabras queden separadas por espacios en blanco, al lograr esto el proceso de tokenización se vuelve más simple (Paute et al., 2016) realizó una clasificación de textos político-electorales mediante la utilización de algoritmos de aprendizaje supervisado con la finalidad de tomar decisiones en tiempo real, allí realizan varias pruebas de implementación una prueba relevante para el caso de estudio del proyecto es la de la implementación del proceso con tokenización y sin tokenización, sin tokens el algoritmo brinda un resultado de 86% de eficiencia en la clasificación mientras que al implementar esta técnica la eficiencia aumenta un 3,3%, es decir, la eficiencia con esta técnica es de 89,3%.

### 1.7.2. Normalización.

Este paso del pre-procesado de un texto es importante porque brinda un estándar en los textos, es decir, los siguientes pasos van a ser procesados uniformemente por lo que en esta etapa se debe convertir todo el texto a mayúsculas o minúsculas, convertir números a sus equivalentes en palabras, eliminar los signos de puntuación, de esta manera (KDnuggets, 2017) define este proceso, además dice que esta técnica permite corregir errores humanos por ejemplo escribir letras de mas (Buenooo) u omitir letras (q en vez de que), mezclarlas, entre otros. Esta etapa se lleva tanto en el stemming o la lematización.

### 1.7.3. Palabras de paro (Stop Words).

(Srividhya & Anitha, 2010) Realizan un análisis del lenguaje español y dice que existen muchas palabras que no brindan información relevante acerca de un tema

en específico, son palabras que se utilizan únicamente para darle una estructura al idioma como lo son las preposiciones, conjunciones, pronombres, artículos, entre otros. Por ejemplo “la”, “de”, “en” las tres son palabras vacías, pero estas son imprescindibles a la hora de establecer una comunicación, en su trabajo realizan un análisis de las diferentes técnicas y combinaciones entre estas, ellos muestran que al eliminar las palabras de paro aumenta el impacto en el rendimiento de manera positiva, también enfatizan en el hecho de la importancia del pre-procesamiento en la clasificación de textos.

### 1.7.4. Stemming.

(Srividhya & Anitha, 2010) Aclaran que esta técnica busca principalmente encontrar la raíz de una palabra, en el idioma español y en todos los lenguajes las palabras se conjugan para expresar tiempos verbales, por ejemplo “Dibujó”, “Dibujará”, “Dibujaría”, las tres palabras están conjugadas y la técnica busca convertir estas palabras en “Dibujar”, el stemming maneja la teoría de que todas las palabras que tienen la misma raíz buscan expresar una idea similar, esta es una buena técnica a utilizar en este proyecto debido a que los tweets y sobre todo en la política utilizan varios tiempos verbales, los actores políticos expresan sus mensajes en tiempo futuro o en algunas ocasiones expresan situaciones que ya ocurrieron, por lo que identificando los verbos y llevándolos a su raíz se identifica cuales tweets están relacionados, en su conclusión dicen que al utilizar las palabras de paro y luego las técnicas de stemming y lematización el rendimiento de la clasificación aumenta notablemente.

### 1.7.5. Lematización.

(Adrián García Diéguez, 2014) define la lematización, al igual que el stemming, como encontrar la raíz de una palabra, la diferencia se encuentra en el idioma a analizar, el stemming se utiliza en idiomas poco flexibles como lo es el inglés, es decir, idiomas en los que no se identifica fácilmente la raíz de la palabra, la lematización es utilizada en idiomas como lo son el español, francés y en general idiomas derivados del latín los cuales son idiomas flexibles, esto quiere decir que están estructurados con una raíz (lexema) y una terminación (morfema), un ejemplo es la palabra “esta + do” donde la primera parte se identifica claramente la raíz “estar” y su complemento, existen 2 tipos de terminaciones en el idioma lo que hace que existan 2 tipos de lematización, la primera llamada lematización flexiva en donde se eliminan las terminaciones verbales, los plurales y el género, la segunda es conocida como la lematización derivativa en donde se eliminan los sufijos, (Adrián García Diéguez, 2014), esta tecnica se puede utilizar en conjunto con las palabras de paro agregando todos los tiempos verbales al diccionario de palabras de paro, tambien resalta que los verbos no brindan información tan importante como la que brinda los sustantivos, adejetivos, entre otros. Esto permite identificar aspectos relevantes y evitar la perdida de informacion.

El procesamiento del lenguaje natural (PLN) es la etapa siguiente al pre procesamiento de textos, mediante del lenguaje natural se puede hacer una

similitud humano-máquina lo que permite analizar y procesar el lenguaje humano con el fin de encontrar patrones, analizar sentimientos, clasificar textos, entre otras diferentes funciones. Las técnicas utilizadas para el PLN para la clasificación de textos cortos en la actualidad son:

### 1.7.6. Etiquetado POS.

(Ritter, Clark, & Etzioni, 2011) dicen que el etiquetado POS (Part Of Speech) consiste en asignar una categoría gramatical (Adjetivo, Verbo, Pronombre, entre otros) a cada una de las palabras de un texto, cada categoría aporta una gran cantidad de información y permite conocer las palabras relacionadas entre sí, la red social twitter maneja un lenguaje informal por lo que en ocasiones existen palabras que no son consideradas gramaticalmente correctas, para solucionar este problema se establecen etiquetas objetivo y el etiquetado POS se encarga de clasificar estas palabras informales en estas etiquetas, esto permite mejorar la eficiencia de la clasificación, “este etiquetado es el primer paso del PLN (Procesamiento del lenguaje natural), de este etiquetado surgen las diferentes técnicas y metodologías del PLN” (Zontisa, 2018) para la clasificación pacifista, guerrerista o neutra del proyecto no es importante considerar estas palabras gramaticalmente incorrectas ya que los actores involucrados son personas de cargos importantes como lo es el presidente y estos están obligados a escribir correctamente porque de esto depende su reputación.

### 1.7.7. Análisis semántico superficial (Shallow Parsing).

Es una técnica conocida también como “chunking” (Ritter et al., 2011) la define como el proceso de etiquetar partes de una oración con roles semánticos tal y como lo hace un etiquetado POS, la principal diferencia entre estas 2 técnicas consiste en que el etiquetado POS clasifica cada palabra en su categoría semántica mientras que el análisis semántico superficial divide la oración en grupos y estos grupos resaltan solo la información con valor para la clasificación. Estos grupos son: frases con sustantivos, verbos y preposiciones (Ritter et al., 2011) en su experimento muestra que al utilizar esta técnica se reduce el porcentaje de error en un 17% además brinda una eficiencia del 86,7% y al combinarlo con otras técnicas la eficiencia puede aumentar considerablemente.

### 1.7.8. Reconocimiento de entidades nombradas (Named entity recognition).

(Ritter et al., 2011) dicen que la metodología de esta técnica es similar a la del análisis semántico superficial pero busca identificar sustantivos relevantes, para este proceso busca identificar 4 categorías (personas, lugares, organizaciones y otros), esto es realizado mediante la segmentación del texto, existen varios esquemas pero los 2 principales son: BIO, este sugiere al clasificador identificar el principio, interior y afuera de los segmentos, el segundo esquema es BILOU el cual sugiere identificar el principio, interior, últimos tokens de cada segmento y longitud de cada segmento, para poder lograr este reconocimiento se utilizan metodologías como HMM (Cadenas ocultas de Markov) y CRF (Campos aleatorios condicionales), esta técnica tiene como finalidad la identificación de sentimientos

asociados a las categorías, por ejemplo a una organización, un lugar (Ratinov & Roth, 2009) o como lo es el caso de este proyecto al proceso de paz.

### 1.7.9. Bolsa de palabras (Bag of words).

(Le & Mikolov, 2014) especifica esta técnica en convertir un texto en un vector el cual cuenta las veces que esta cada palabra en el texto, después de convertido permite muchas aplicaciones, pero para el caso de estudio trabajado en este documento, esta técnica simple no sirve ya que se pierde el orden de las palabras lo cual reduce drásticamente la precisión en la clasificación, para darle solución a este inconveniente surgen los bigramas y los n-gramas los cuales están estructurados de manera que la palabra siguiente está ligada con la anterior, un ejemplo de esto sería: “Desde que iniciamos el acuerdo de paz …” una estructura de bigrama seria: “Desde: que” “que: iniciamos” “iniciamos: el” y así sucesivamente, esto incrementa cierto porcentaje de clasificación pero la semántica se sigue perdiendo.(Le & Mikolov, 2014), en el análisis de los resultados de su trabajo se evidencio que la técnica tiene un error en la clasificación del 11.77%, también existe una combinación de técnicas que puede llegar a reducir el error hasta un 10.77%, esto deja una eficiencia del 89,3% siendo un porcentaje interesante.

### 1.7.10. Palabras embebidas (Word embedding).

Al igual que la bolsa de palabras, (Levy, Goldberg, & Dagan, 2015) dicen que el Word embedding busca convertir un texto en un vector numérico, cada tweet se representa en un vector numérico y todos estos vectores se localizan en un espacio dimensional y se basan en el supuesto que los vectores que estén más cerca son semánticamente relacionados, (Kusner, Sun, Kolkin, & Weinberger, 2015) dicen que este algoritmo ha venido evolucionado hasta que surgió el algoritmo word2vec el cual es usado en la actualidad, de esta técnica parte la aplicación de bolsa de palabras y la de TF-IDF (Frecuencia de termino-frecuencia de documento inverso), ellos hablan de que el TF-IDF proporciona buenos resultados, además de esto es la única técnica con esas características que proporciona buenos resultados, la bolsa de palabras también proporciona resultados interesantes.

Existen varios algoritmos de aprendizaje supervisado y técnicas probabilísticas las cuales son entrenadas con un grupo de datos históricos que a futuro permite la clasificación de textos y realiza predicciones permitiendo solucionar un problema específico, algunos de los algoritmos más utilizados para la clasificación de textos cortos son:

### 1.7.11. Máquinas de soporte vectorial (SVM).

(Santana Mansilla, Costaguta, & Missio, 2014) define este clasificador consiste en una serie de algoritmos el cual recibe vectores numéricos los cuales son formados con los textos de entrenamiento y el SVM construye una serie de híper-planos, estos híper-planos son construidos mediante vectores de soporte los cuales son

contenidos en un espacio de n dimensiones, estos híper-planos constan de 2 partes, la positiva y la negativa, entre estas partes va a existir un margen que las separa, es decir, el margen va a ser el punto más cercano a los 2 conjuntos, (Reyes-Ortiz, Paniagua-Reyes, & Sánchez, 2017). Al ingresar un texto nuevo, el clasificador puede predecir la clase a la que pertenece, para el caso de estudio del proyecto la utilización de las máquinas de soporte vectorial permiten predecir la orientación pacifista, guerrerista o neutra de cada tweet según la clase más cercana del texto ingresado, (Reyes-Ortiz et al., 2017) implementan el algoritmo de SVM y obtiene en uno de sus ejemplos una eficiencia máxima de 89% estas eficiencias oscilan entre el 79-89% por lo que es importante tener en cuenta la estructura de los datos a la hora de implementar este algoritmo.

### 1.7.12. KNN.

(Barve, Rahate, Gaikwad, & Patil, 2018) especifica que este clasificador es basado en un algoritmo de aprendizaje supervisado el cual tiene 2 fases: el proceso de entrenamiento en el cual recibe un conjunto de vectores con las respectivas etiquetas y los vectores son ubicados en un espacio multidimensional, la segunda fase es el proceso de clasificación en donde el algoritmo busca el o los puntos más cercanos, la clase que tenga el punto o los puntos más cercanos será la clase a la que pertenecerá el texto.

(Cambronero & Moreno, 2010) dice que este clasificador cuenta con 2 reglas: la primera regla es la más simple y es llamada 1-NN, esta es basada en la suposición en la cual el texto a clasificar va a pertenecer clase más cercana que encuentre por lo que el algoritmo evaluará la distancia entre puntos y la de menor distancia es la que elegirá, la segunda regla es la regla general, en donde se elige un rango k el cual delimita un espacio mediante un circulo, para un espacio bidimensional y una esfera para un espacio multidimensional y evalúa la cantidad de puntos existentes dentro del rango, para la elección de un k optimo se debe tratar de abarcar gran parte en donde el área este poblada, la mayor cantidad de puntos que encuentre va a ser la clase a la que pertenecerá el texto (Cambronero & Moreno, 2010) dice que el algoritmo es robusto debido a que es tolerante al ruido lo que permite preocuparse menos por la estructura de los datos, pero cuenta con el problema de que si el número de descriptores aumenta el rendimiento del mismo tiende a reducir.

(Barve et al., 2018) realizaron una clasificación de mensajes de alerta de ataque terroristas, estos mensajes son enviados al celular por lo que son considerados textos cortos, en su trabajo utilizaron 3 clasificadores, KNN, SVM y Random Forest, al ver los resultados se puede evidenciar que el algoritmo que brindó mejores resultados para el caso de estudio fue el KNN seguido de SVM y por último el Random Forest.

### 1.7.13. Arboles de decisión.

(Del Pilar & Robles, 2017) plantea que el objetivo del algoritmo es construir un diagrama en forma de árbol el cual permita hacer un seguimiento a cada uno de los datos del conjunto de entrenamiento, para la utilización de este algoritmo se debe comenzar eligiendo un valor de entrada, a este se le evalúa una característica y dependiendo de su valor se elige a uno de sus nodos hijos y se evalúa otra característica, este funcionamiento del algoritmo continua hasta que llegue hasta las hojas (etiquetas de clasificación) y enviará la clasificación de la muestra de entrada que se seleccionó con su respectivo camino elegido (Escortell Pérez, Giménez Fayos, & Rosso, 2017) hablan que este algoritmo es usualmente utilizado en problemas donde la función objetivo tiene valores discretos, cuando se tienen expresiones separadas, los datos de entrenamiento pueden tener errores o cuando se requiere una creación de descripciones dice (Del Pilar & Robles, 2017).

(Bifet & Frank, 2010) realizaron un análisis de sentimientos a publicaciones de la red social twitter con el fin de establecer una polaridad positiva o negativa, para esto aplicaron 3 algoritmos: Naive Bayes, SGD y Arboles de decisión, se realizaron varias pruebas con diferentes set de datos y se evidenció que los Arboles de decisión tuvieron los peores resultados con una eficiencia de entre 69 y 73 % mientras que los otros 2 algoritmos tuvieron eficiencias entre 75 y 83 %.

### 1.7.14. Naive Bayes.

Este algoritmo es fundamentado por el teorema de Bayes (Valdivia, Eugenio Martínez Cámara, M. Teresa Martín Valdivia, José M. Perea Ortega L, 2011) dicen que lo que busca es hacer una estimación de la probabilidad de que un documento pertenezca a cierta categoría (Figuerola, Alonso Berrocal, Zazo Rodríguez, & Rodríguez, 2004) afirma que este clasificador, al ser de metodología supervisada, se le debe proporcionar un conjunto de datos de entrenamiento, luego procesa este conjunto y al hallar la frecuencia y combinaciones estima las probabilidades de ocurrencia (Patil & Sherekar, 2013) dice que para que este algoritmo sea efectivo se le necesita proporcionar un grupo de datos de entrenamiento (Valdivia, Eugenio Martínez Cámara, M. Teresa Martín Valdivia, José M. Perea Ortega L, 2011) realizan una clasificación utilizando diferentes técnicas de pre-procesamiento, la eficiencia del algoritmo varía entre el 62 al 84%, la técnica que se usa en el porcentaje más alto es TF-IDF con stop words, (Bifet & Frank, 2010) realizaron pruebas con este algoritmo en la clasificación de polaridad de tweets y tuvo una eficiencia del 75 al 82% aplicando el algoritmo sin ninguna técnica de pre-procesamiento,

### 1.7.15. Regresión logística.

(Hosmer & Lemeshow Stanley, 1989) en su texto definen la regresión logística como una técnica basada en probabilidades la cual consta de 2 tipos de variables, la dependiente y la independiente, la variable dependiente tiene como característica que es dicotómica (si o no) y lo que busca este modelo es predecir

cuál de las opciones de la variable dicotómica va a suceder en un instante de tiempo partiendo de la ocurrencia de variables independientes o predictoras. (Xiang, Fan, Wang, Hong, & Rose, 2012) realizan una detección de tweets ofensivos en un set de datos compuesto por 4029 publicaciones en donde este algoritmo logró un 75,1% de eficiencia.

### 1.7.16. Redes Bayesianas.

(Sucar, 2011) dice que estas redes buscan modelar eventos específicos, estos son representados en manera de grafo y lo hacen mediante la identificación de variables y las dependencias que existen entre ellas, las variables son representadas mediante nodos y los arcos representan la relación de dependencia directa entre variables, para poder realizar una predicción estas redes utilizan la inferencia bayesiana la cual consiste en estimar probabilidades de ocurrencia de las variables no conocidas en base a las variables conocidas, esta técnica es muy utilizada en problemas de clasificación y predicción (Sucar, 2011) considera una buena técnica robusta.

### 1.7.17. Cadenas ocultas de Markov.

(Shashanka, 2011) plantea estas cadenas ocultas las cuales modelan una secuencia de datos observados, este modelado asume que la estructura es un modelo de Markov y genera una serie de estados ocultos, luego calcula las probabilidades de transición entre estados para calcular las probabilidades de estados ocultos, de estas cadenas ocultas se derivan varios algoritmos como el forward-backward o el de baum-welch o Viterbi, el algoritmo de Viterbi el cual consiste en conocer la secuencia de estados más probable dada una secuencia de estados observables, (Tornero Lucas, 2017) dice que estos estados tienen como característica que no es la primera secuencia de estados encontrados si no que es la secuencia de estados más creíble, más verosímil, para garantizar esto utiliza una variable auxiliar que almacena la probabilidad más alta de una secuencia de estados, si encuentra otra probabilidad más alta en otra secuencia de estados se modifica la variable.

### 1.7.18. Redes Neuronales.

(Tornero Lucas, 2017) dice que las redes neuronales son una evolución de los árboles de decisión, de igual manera que los arboles su estructura contiene nodos los cuales son llamados neuronas y estas están conectados por enlaces y cada uno tiene un peso asociado, también tiene nodos ocultos los cuales permiten la interacción no lineal con las variables de entrada, cuando se entrena esta red se modifican todos los pesos y esta se convierte en una red con un rendimiento robusto al encontrarse con patrones de entrada ruidosos o incompletos, alta tolerancia a fallos y la capacidad de clasificar y predecir (Dong et al., 2014) utilizan una red neuronal para clasificar sentimientos de tweets, esta red dio un resultado de baja precisión, alrededor de un 65% de eficiencia, sin embargo un equipo que quedo ubicado en el top 20 del concurso (Rosenthal, Farra, & Nakov, 2017) utilizo redes neuronales para publicaciones en lengua arábica.

## 1.8. METODOLOGÍA.

Se implementa la metodología de desarrollo ágil XP, esta metodología de desarrollo es escogida por ser ágil la cual permite cambios durante la ejecución del proyecto, también es escogida por la cantidad integrantes del proyecto “dos”, que según (Universidad Nacional Mayor de San Marcos. Facultad de Ingeniería Industrial. Instituto de Investigación, Oscar; Rosales López, Pedro Pablo; Salas Bacalla, 2010), son dos parámetros importantes a la hora de seleccionar una metodología de desarrollo, también se debe tener en cuenta el criterio de (Letelier & Penadés, 2006) en donde explica que la retroalimentación continua es importante en el desarrollo de la metodología seleccionada y la comunicación continua entre el cliente y el equipo de trabajo.

Se desarrolla las tres características esenciales de la metodología seleccionada, como lo son:

- Historias de usuario.
- Roles.
- Proceso y prácticas.

### 1.8.1. Historias de usuario.

Son relatos en el cual se detalla la idea del cliente, estas las analiza el equipo de trabajo para establecer los requerimientos funcionales y no funcionales. Las historias de usuario pueden contener uno o varios requerimientos ya sean funcionales y no funcionales y este criterio va directamente en el equipo de desarrollo. Una historia de usuaria debe ser detallada y comprensible, para que así el programador pueda implementarla en poco tiempo (semanas o días). Es importante que exista al menos una historia de usuario por cada característica importante del sistema, la cantidad de historias puede ir aumentando y depende de las reuniones que se realicen con el cliente para seguimiento y evaluación de los entregables. (Letelier & Penadés, 2006).

### 1.8.2. Roles.

Se define como el equipo de trabajo a los siguientes integrantes: Cesar Espitia y Juan Pablo Paramo y se definen los siguientes roles.

- Desarrollador: El equipo de trabajo está a cargo de la producción del código del sistema y las pruebas unitarias.
- Administrador de Base de datos (NoSQL – Relacionales): Cesar Espitia es el encargado administrar los datos contenidos en la base de datos necesaria para la resolución de proyecto.

- Encargado pruebas: Juan Paramo realiza las pruebas funcionales las cuales se ejecutan de manera regular, también selecciona la herramienta a utilizar para realizar las pruebas mencionadas.
- Encargado de seguimiento (Tracker). El equipo de trabajo realiza el seguimiento a cada iteración y también verifica el grado de acierto entre las estimaciones realizadas y el tiempo dedicado.
- Entrenador (Coach): El equipo de trabajo entreno el algoritmo bajo el concepto de Machine Learning.

### 1.8.3. Proceso.

Dentro del ciclo de vida a seguir para el desarrollo del proyecto, se seguirán las siguientes fases.

1. Exploración.
2. Planificación de la entrega (Release).
3. Iteraciones.
4. Producción.
5. Mantenimiento.
6. Muerte del Proyecto.

#### 1.8.3.1. Exploración.

En esta fase el equipo de trabajo define las historias de usuarios para la construcción de los requerimientos funcionales y no funcionales, el equipo de trabajo comienza a trabajar sobre la(s) herramienta(s) seleccionada(s) utilizando los algoritmos seleccionados.

#### 1.8.3.2. Planificación de la entrega.

El equipo de trabajo define las historias de usuarios que se desarrollan para la entrega, de igual manera define la prioridad de cada uno de ellas y la fecha de estimación para la entrega de cada una.

#### 1.8.3.3. Iteraciones.

Las iteraciones o ciclo de desarrollo, dependen de la prioridad de la historia de usuario a desarrollar, en la cual se tratan las que tengan mayor prioridad. La iteración tendrá como un máximo de tiempo para trabajar en ella de una semana.

#### 1.8.3.4. Producción.

Antes de realizar la entrega formal, se llevan a cabo las pruebas y se elaboran las conclusiones del proyecto.

1.8.3.5. Mantenimiento.

Una vez entregado y sustentado el proyecto no se tiene contemplado realizar ningún mantenimiento sobre el mismo.

1.8.3.6. Muerte del Proyecto

Cuando se realice la entrega del proyecto, se terminará su desarrollo y no se admitirán nuevos cambios sobre el mismo.

## 1.9. PRESUPUESTO

En el presupuesto para realizar el proyecto de grado se están estimando las siguientes variables.

Tabla 1. Presupuesto

| | 1er Mes | 2do Mes | 3er Mes | 4to Mes | 5to Mes | 6to Mes | 7mo Mes | 8vo Mes | 9no Mes | 10mo Mes | 11avo Mes | 12avo Mes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Horas de trabajo individual | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 | 60 |
| Costo horas de trabajo | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 | $780.000 |
| Internet | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 | $69.990 |
| Energia | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 | $46.463 |
| Equipo de computo | $3.820.046 | | | | | | | | | | | |
| Costo | $4.716.499 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 | $896.453 |

Fuente: Autores.

Donde el costo de las horas de trabajo está calculado de la siguiente manera, se multiplica la cantidad de horas asignadas por el número de ingenieros que trabajan en el proyecto, el resultado se multiplicara por la hora de trabajo que tiene un valor de $6.500 COP que da un resultado de $780.000 COP.

El costo de internet está calculado con una velocidad de 5 Megas de navegación para un estrato tres que, según los proveedores del servicio, esto está estimado en $69.990 COP.

De igual manera se calcula el valor de la energía, para un estrato 3 se estima un consumo medio de 124 (kWh/mes) por un valor de $46.463.

Y por último se escogieron dos equipos de cómputo, cada uno con un valor de $1.910.023 de marca Toshiba con un modelo Tecra C40-D1412.

Tabla 2. Costo total del proyecto.

| | |
|---|---|
| Costo total del proyecto | 14.577.482 |

Fuente: Autores.

## 1.10. PRODUCTOS A ENTREGAR

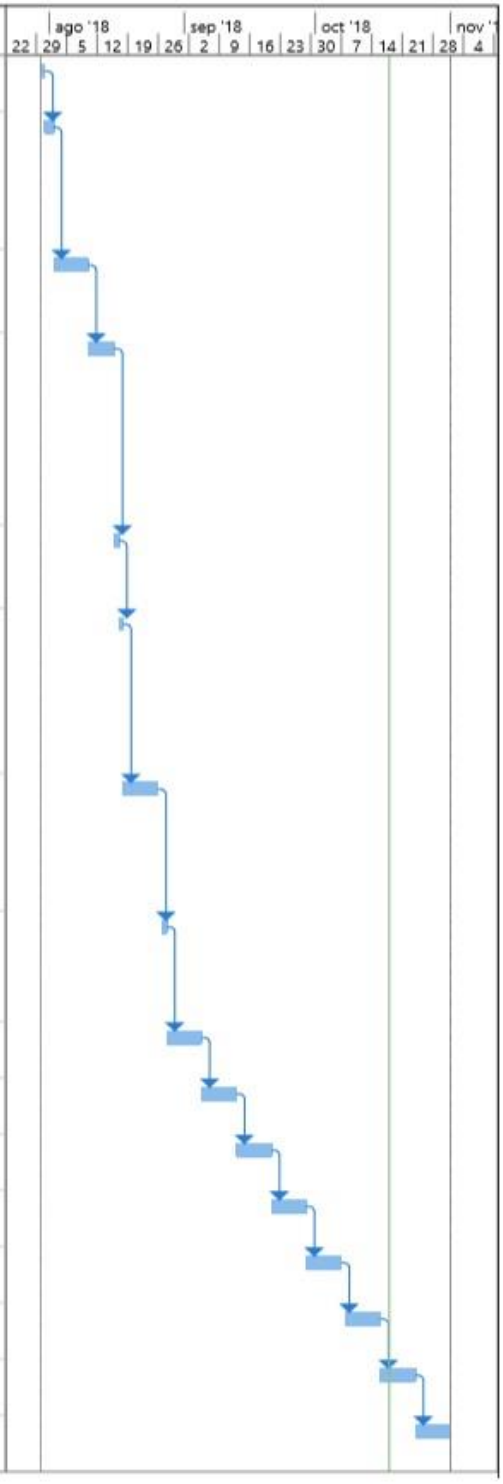
Se tiene estimado realizar la entrega de:

- Algoritmos elegidos y tabla de eficiencia después de la elaboración de conclusiones.
- Manual de usuario.
- Manual de programador.
- Código (script) generado con la plataforma, paquete, entorno o arquitectura seleccionada y que utilice los algoritmos seleccionados.

## 1.11. CRONOGRAMA

Ilustración 2. Cronograma de actividades.

| Id | Modo de tarea | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
|---|---|---|---|---|---|---|
| 1 | | Introducción al proyecto de grado | 1 día | lun 30/07/18 | lun 30/07/18 | |
| 2 | | Aclaración de dudas en el proyecto de grado y elaboración de los objetivos generales y específicos | 2 días | mar 31/07/18 | mié 1/08/18 | 1 |
| 3 | | Investigación y construcción de la bibliografía | 7 días | jue 2/08/18 | jue 9/08/18 | 2 |
| 4 | | Elaboración de la introducción, descripción del problema y pregunta científica orientada a la formulación del problema | 5 días | vie 10/08/18 | mié 15/08/18 | 3 |
| 5 | | Validación de los Objetivos generales y específicos | 1 día | jue 16/08/18 | jue 16/08/18 | 4 |
| 6 | | Creación de la justificación del proyecto al igual que la delimitación (Espacio, tiempo, contenido y alcance) | 1 día | vie 17/08/18 | vie 17/08/18 | 5 |
| 7 | | Elaboración del Marco referencial (Conceptual, teórico, geográfico, demográfico) y el Estado del arte | 7 días | sáb 18/08/18 | sáb 25/08/18 | 6 |
| 8 | | Implementación de la metodología y construcción de las historias de usuario | 1 día | lun 27/08/18 | lun 27/08/18 | 7 |
| 9 | | Ejecución de la primera historia de usuario | 7 días | mar 28/08/18 | mar 4/09/18 | 8 |
| 10 | | Ejecución de la segunda historia de usuario | 7 días | mié 5/09/18 | mié 12/09/18 | 9 |
| 11 | | Ejecución de la tercera historia de usuario | 7 días | jue 13/09/18 | jue 20/09/18 | 10 |
| 12 | | Ejecución de la cuarta historia de usuario | 7 días | vie 21/09/18 | vie 28/09/18 | 11 |
| 13 | | Ejecución de la quinta historia de usuario | 7 días | sáb 29/09/18 | sáb 6/10/18 | 12 |
| 14 | | Ejecución de la sexta historia de usuario | 7 días | lun 8/10/18 | lun 15/10/18 | 13 |
| 15 | | Ejecución de la séptima historia de usuario | 7 días | mar 16/10/18 | mar 23/10/18 | 14 |
| 16 | | Ejecución de la octava historia de usuario | 7 días | mié 24/10/18 | mié 31/10/18 | 15 |




Fuente: Autores.

# 2. PLATAFORMAS Y ENTORNOS

## 2.1. PLATAFORMAS

Se realiza la investigación pertinente para 3 plataformas las cuales están enfocadas en la aplicación y desarrollo del presente proyecto, en cada una de ellas se han realizado la implementación y desarrollo de proyectos similares.

### 2.1.1. WEKA.

Es una plataforma basada en Java, surge en Nueva Zelanda en la universidad de Waikato con la filosofía de software libre, puede ser utilizada en diferentes sistemas operativos como lo son: Windows y Linux. Esta plataforma está compuesta por técnicas de pre-procesamiento de textos, procesamiento del lenguaje natural y algoritmos de machine learning, también incluye métodos que permiten solucionar problemas de minería de datos como lo son: regresión, clasificación, clustering, entre otros. Es una herramienta completa y su interfaz gráfica es bastante intuitiva por lo que es de fácil adaptación para el usuario, su interfaz está compuesta por 4 elementos, el explorador el cual permite fácilmente cargar un set de datos y aplicar un algoritmo de aprendizaje arrojando varios resultados que se deben interpretar, existe el Knowledge Flow el cual permite arrastrar elementos lo que permite entender como es el ciclo de los datos, el experimentador el cual permite analizar diferentes algoritmos con diferentes parámetros ayudando a la toma de decisiones, por ultimo está el workbench el cual integra las 3 anteriores, al estar desarrollada en java le permite fácil conexión con bases de datos tanto estructuradas como no estructuradas (Frank, Hall, Witten, & Kaufmann, 2016a).

Weka cuenta con un formato de archivos con extensión arff el cual tiene una estructura simple: consta de una etiqueta @RELATION la cual define el nombre del dataset, con @ATTRIBUTE donde se definen los atributos que contiene cada fila del dataset, estos atributos pueden ser de cualquier tipo (numéricos, cadenas, caracteres, entre otros) y la etiqueta @DATA en donde le dice a weka donde están localizados los datos para poder trabajar con estos, es importante resaltar que los atributos están separados por comas por lo que los atributos de tipo String es recomendado ponerlos entre comillas simples o dobles, además permite trabajar con archivos Excel y otra gran cantidad de tipos de archivos (Frank, Hall, Witten, & Kaufmann, 2016b).

### 2.1.2. Rapid Miner Studio.

Esta plataforma fue creada por la empresa Rapid Miner INC, está diseñada en el lenguaje Java enfocada al machine learning por lo que cuenta con una librería extensa de algoritmos de aprendizaje automático, cuenta con un componente de Deep learning, al igual que Weka cuenta con conexión a bases de datos estructuradas y no estructuradas, pero también permite trabajar con varios tipos de archivos por lo que el usuario no se debe preocupar en convertir un dataset a un formato especifico, cuenta con una interfaz gráfica intuitiva, tutoriales, comunidad activa, documentación, lo que le permite al usuario apropiar la herramienta en poco tiempo, a diferencia de Weka, Rapid Miner no tiene el pensamiento de software libre por lo que las herramientas que brinda en la versión gratuita son básicas, brinda ciertos algoritmos los cueles tienden a ser pocos, existe una herramienta de la plataforma llamada “Auto Modelling” la cual en una serie de pasos va modelando la secuencia por lo que es una gran manera de ahorrar tiempo (RapidMiner, 2014). Sin embargo, solo está disponible en la versión de pago, permite el procesamiento de data sets pequeños, resumiendo un poco es una gran herramienta si se realiza el pago.

Este software está diseñado de manera en que el usuario entienda el flujo de datos, es decir, está estructurado de manera en que son arrastrados los componentes y permite al usuario entender desde cómo se carga el set de datos, ver las modificaciones que se le va haciendo a lo largo de cada proceso, como se va modificando tanto en las etapas de pre-procesamiento, procesamiento del lenguaje natural, en el algoritmo de aprendizaje, la herramienta también ayuda a la experiencia de usuario ya que si el usuario no tiene conocimiento de un problema y no sabe cómo solucionarlo en ocasiones esta herramienta tiene una ayuda la cual en casos puede agregar elementos faltantes o establecer parámetros que el usuario ha olvidado, la plataforma tiene una buena manera de visualizar los datos, cuenta con varios tipos de graficas que permite entender como están distribuidos los datos y permite tomar decisiones acertadas.

### 2.1.3. KNIME.

Es una plataforma de análisis de datos, que permite la integración abierta y la presentación de informes, se desarrolló originalmente en el departamento de bioinformática y minería de datos de la Universidad de Constanza, Alemania. Es una plataforma de código abierto escrita en Java con licencia GPL, esta plataforma permite mediante un entorno visual desarrollar modelos de minería de datos, se puede utilizar en cualquier sistema operativo como los son Windows, Linux, Mac OSX y cualquier S.O. que soporte máquina virtual JAVA.

Dentro de sus funcionalidades permite la integración de múltiples herramientas al igual que permite trabajar con datos de diferentes fuentes, es una herramienta intuitiva con un fácil manejo y fácil aprendizaje en su utilización. Esta herramienta

es normalmente utilizada para el análisis de datos de un cliente (CRM), análisis financiero e inteligencia de negocios.

Esta plataforma permite crear visualmente un flujo de datos, ejecuta selectivamente todas o algunas etapas en el análisis, al final de este proceso muestra a través de modelos y vistas los resultados de tal manera que permite su investigación, entre las vistas están incluidos diagramas de dispersión, coordenadas paralelas entre otros, se especializa en el Business Intelligence, minería de datos y para la presentación de informes empresariales (Rangra & Bansal Research Scholar Professor, 2014).

Tabla 3. Ventajas y desventajas de utilizar plataformas

| | Ventajas | Desventajas |
|---|---|---|
| Weka | • Software libre, con soporte y documentación.<br>• Se encuentra en constante desarrollo.<br>• Interfaz gráfica intuitiva.<br>• Gran cantidad de técnicas de pre-procesamiento, procesamiento del lenguaje natural y algoritmos de aprendizaje supervisado y no supervisado. | • Visualización de datos en gráficos pobre.<br>• Solo sirve como herramienta de inteligencia de negocios.<br>• Permite pequeños y medianos sets de datos. |
| Rapid miner | • Combina minería de datos, con minería de texto, aprendizaje de máquina, Inteligencia de negocios y análisis de negocios.<br>• Buenos resultados en gráfica.<br>• Incluye varios de los algoritmos implementados en weka. | • Ofrece unas características libres pero a partir de un punto son de pago por lo que los algoritmos y algunas características son limitados.<br>• Requiere gran capacidad de memoria RAM. |
| KNIME | • Interfaz gráfica intuitiva.<br>• No necesita instalación, se puede usar desde la carpeta raíz.<br>• Permite el análisis predictivo<br>• Interactúa con programas que permiten la visualización y análisis de datos. | • Necesita un equipo con buena capacidad de memora RAM.<br>• No tiene métodos para la selección de descriptores.<br>• No tiene facilidad automática para la optimización de parámetros de aprendizaje automático y de métodos estadísticos. |

Fuente: Autores.

## 2.2. ENTORNOS

Se realiza la investigación pertinente para 3 entornos de desarrollo o lenguajes de programación, los cuales están enfocados en la aplicación y desarrollo del presente proyecto, en estos entornos se ha realizado la implementación y desarrollo de proyectos similares por lo cual son una buena opción para desarrollar en el presente proyecto.

### 2.2.1. MATLAB.

Es una herramienta que contiene su propio lenguaje de programación, se conoce como un laboratorio de matrices y es diseñado para realizar cálculos técnicos. Se puede utilizar en S.O, Mac OS X, GNU/Linux, Unix y Windows. MATLAB se puede aplicar para diseño de sistemas de control, procesamiento de señales, identificación de sistemas, redes neuronales, simulación de sistemas dinámicos y otros, integrando el cálculo la programación y la visualización en una notación matemática, se especializa en la resolución de problemas que incluyan vectores y matrices, dentro de sus principales ventajas esta las toolboxes las cuales cuentan con documentación, normalmente es utilizado para analítica de datos, comunicaciones inalámbricas, machine learning, visión artificial, robótica, sistemas de control y finanzas cuantitativas y gestión de riesgos.(The MathWorks Inc, 2018)

### 2.2.2. Python (Scikit).

Es una librería para el lenguaje de programación Python, al ser utilizada en Python la librería puede incluirse en sistemas operativos como lo son Windows y Linux, está documentada y cuenta con foros y comunidades que permite solucionar problemas rápido, además de esto cuenta con soporte, existen varias empresas de talla mundial que utilizan esta librería como lo son: Spotify, Evernote, booking.com, entre otros. Scikit esta es especializada en machine learning por lo que cuenta con gran variedad de algoritmos de aprendizaje automático disponibles, también Python cuenta con librerías para el pre-procesamiento, procesamiento del lenguaje natural, visualización de datos por lo que con las librerías indicadas se convierte en un lenguaje muy potente para el machine learning (*scikit-learn user guide*, 2018).

Cuenta con tareas de clasificación en donde se identifica la clase a la que un objeto pertenece, regresión que predice atributos relacionados a un objeto, clustering que agrupa objetos similares en diferentes sets, reducción dimensional que permite reducir el número de variables aleatorias a considerar en el proceso de clasificación y regresión, pre-procesamiento en donde se realiza una extracción de características y normalización, agregándole a esto cuenta con bastantes ejemplos en donde se muestra el paso a paso de cada una de las características que posee y a que situaciones puede ser utilizada como lo son: el reconocimiento

de imágenes, de voz, de texto, (scikit learn, 2017), por lo que se puede decir que es una herramienta potente y completa.

### 2.2.3. R.

Es un lenguaje de programación distribuido con licencia GNU el cual puede compilar en diferentes sistemas operativos como lo son Windows y Linux, R está basado en brindar modelos estadísticos y creación de graficas de los mismos, también cuenta con la posibilidad de agregar librerías, para el caso de machine learning se utiliza la librería llamada “caret” (Venables & Smith, 1997) la cual cuenta con varias técnicas de pre-procesamiento, división del conjunto de datos, entrenamiento del modelo, predicciones, clustering, medición del rendimiento, algoritmos genéticos, entre otros (Kuhn, 2018).

Al tener como foco la estadística la manera de graficar los datos es excelente, tiene distintas formas de gráficas, tanto en 2d y 3d, de puntos, de caja, funciones de densidad, curvas, todas estas permiten entender como están distribuidos los datos (Kuhn, 2018), permitiendo al analista decidir qué modelo será el más eficiente brindando los mejores resultados, el problema de tener tan buenas graficas es que el tiempo de procesamiento es más alto por lo que se requiere de buenos equipos con alta memoria permitiendo compilar y visualizar las gráficas.

Tabla 4. Ventajas y desventajas de utilizar lenguajes de programación

| | Ventajas | Desventajas |
|---|---|---|
| Python | • Sintaxis simple, fácil de utilizar.<br>• Tiempo de compilación rápido.<br>• En constante mejora. | • Brinda funcionalidad básica en estadísticas.<br>• Limitaciones para acceso a bases de datos.<br>• Difícil de leer los set de datos (en comparación a las plataformas) |
| R | • Buena forma de visualización de datos.<br>• Excelente Vectorización.<br>• Facilidad para importar y exportar datos. | • Tiempo de compilación lento.<br>• Problemas con la memoria.<br>• Los paquetes de ML son muy diferentes unos de otros, no hay un estándar y no es especializado en el área. |

| | | |
|---|---|---|
| MATLAB | • Software con soporte y documentación<br>• Incorporación de toolboxes.<br>• Buen rendimiento al interactuar con matrices y vectores.<br>• Utilización de apps para la programación personalizada.<br>• Confiabilidad en los resultados. | • Presenta los datos binarios como uint8, por ejemplo, en un fichero tipo “mapa de bits”<br>• Para arquitecturas de 64 bits el tamaño máximo del bloque es de 256 TB y para las de 32 el tamaño máximo de 2GB.<br>• Para grandes problemas presenta fallas en rendimiento.<br>• No es fácil utilizar las herramientas de debugging y profiling.<br>• Ofrece una licencia de prueba y de estudiante en donde su utilización es limitada, para acceder a la versión completa hay que adquirir la versión comercial. |

Fuente: Autores.

Estudiando las plataformas y lenguajes de programación que sean aptos para el desarrollo del proyecto, es escogida la plataforma WEKA principalmente por la documentación que posee para su utilización y la aplicación a casos de estudio similares lo cual es de gran ayuda en la resolución del presente proyecto, también es escogida por su facilidad en su utilización ya que es una herramienta intuitiva, además la integración que posee para interactuar con bases de datos es buena y por ser un software libre que posee varios métodos y algoritmos de clasificación que permiten un análisis predictivo, se valida también Matlab y Rapid Miner son de pago por lo que son descartados ya que no se ajustan con nuestro proyecto, los lenguajes de programación como Python y R son una buena alternativa ya que cuentan con buenas herramientas, sin embargo no se cuenta con el tiempo adecuado para poder implementar el objetivo del estudio por lo que también son descartados, finalmente se descarta KNIME por su limitada cantidad de algoritmos implementados.

# 3. ALGORITMOS

En el caso de estudio se estudian los algoritmos con los cuales se podrá realizar clasificación de texto corto, dentro de los cuales se encuentran los siguientes con las respectivas ventajas y desventajas que se obtienen al utilizar uno de ellos:

Tabla 5. Algoritmos con su descripción.

| Algoritmo | Ventajas | Desventajas | Que hace |
|---|---|---|---|
| KNN | Fácil de implementar. | • Difícil encontrar un k óptimo.<br>• Procesamiento lento. | Recibe un conjunto de vectores con las respectivas etiquetas y estos son ubicados en un espacio multidimensional, se elige un rango k, y evalúa la cantidad de puntos existentes dentro del rango, la mayor cantidad de puntos que encuentre va a ser la clase a la que pertenecerá el texto . |
| SVM | • Clasifica bastante bien problemas de variables dependientes.<br>• Se puede aplicar a grandes conjuntos de datos complejos con ruido.<br>• Se pueden utilizar para resolver tanto problemas lineales como no lineales. | • Entrenar con un conjunto de datos demasiado grande resulta ineficiente.<br>• Se puede presentar “overtraining”. | Recibe los datos de entrenamiento y los coloca en un espacio bidimensional, luego encuentra uno o varios híper-planos que permitan separar las clases. Un hiperplano optimo es el que maximiza las distancias entre clases, los puntos usados para definir el hiperplano se llaman vectores de soporte y dependiendo del lado del hiperplano será la clase a la que pertenece el texto. |

| | | | |
|---|---|---|---|
| Naive Bayes | • No es sensitivo a características poco relevantes.<br>• Sus resultados son confiables.<br>• Solo se requiere una pequeña cantidad de datos de entrenamiento para la clasificación. | • No es apto para el manejo de variables aleatorias continuas.<br>• No se pueden modelar dependencias. | Es una técnica que construye modelos para predecir la probabilidad de posibles resultados, teniendo en cuenta el teorema de Bayes el cual consiste en: La probabilidad de que se dé un suceso, habiendo sucedido otro que influye en el anterior. Esta técnica utiliza estos datos históricos para encontrar asociaciones, relaciones y así hacer predicciones. |
| Arboles de decisión | • Se comportan bien en espacios multidimensionales.<br>• Plantean el problema para que todas las opciones sean analizadas.<br>• Facilita la interpretación de la decisión adoptada. | • No es fácil de entrenar.<br>• Pierde el control a medida que se despliega. | A partir de ejemplos conformados como un conjunto de pares atributo–valor arma un esquema en forma de árbol, luego elige un valor de entrada, a este se le evalúa una característica y dependiendo de su valor se elegirá a uno de sus nodos hijos y se evaluará otra característica, este funcionamiento del algoritmo continuará hasta que llegue hasta las hojas (etiquetas de clasificación) y allí enviará la clasificación de la muestra de entrada que se eligió, lo que permite realizar predicciones. |

| | | | |
|---|---|---|---|
| Redes neuronales | • Robustez.<br>• Buena tasa de acierto.<br>• Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.<br>• Tolerancia a fallos: Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente. | • Toman mucho tiempo para entrenarlas.<br>• En la etapa de entrenamiento exigen una elevada cantidad de datos.<br>• Tienden a ser complejas de entender.<br>• Complejidad de aprendizaje para grandes tareas. | Se compone de varias unidades o “neuronas”, inspirados en la naturaleza biológica de las neuronas ya que estas están fuertemente interconectadas, cada neurona contiene palabras y estas tienen un peso. Los pesos en la red son modificados con el fin de reducir el margen de error en la clasificación del texto estudiado. Estas aprenden según los parámetros de entrada y de esta manera, al evaluar un texto nuevo devuelve la clase a la que pertenece. |
| Redes bayesianas | • Fácil de entender.<br>• Es muy robusto considerando atributos irrelevantes.<br>• Toma evidencia de muchos atributos para realizar la predicción final.<br>• Pueden modelar sistemas complejos | • Seleccionar una apropiada distribución de los datos tiene un importante efecto en la calidad de los resultados.<br>• Depende de que el dataset este correctamente seleccionado. | Reciben un conjunto de variables y hallan las relaciones de dependencia entre ellas. Dado este modelo, se puede hacer inferencia bayesiana; es decir, estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas. Estas son representadas por medio de grafos. Como característica tiene que es grafo a cíclico dirigido en el que cada nodo representa una variable aleatoria que tiene asociada una función de probabilidad condicional. |

| | | | |
|---|---|---|---|
| Regresión logística | • Facilidad computacional.<br>• Permite el manejo de múltiples variables independientes. | Necesita que los conjuntos sean linealmente separables. | Consiste en obtener una función logística de las variables independientes que permita clasificar a los individuos en una de las dos subpoblaciones o grupos establecidos por los dos valores de la variable dependiente. La función logística es aquella que halla, para cada individuo según los valores de una serie de variables (Xi), la probabilidad (p) de que presente el efecto estudiado. Una transformación logarítmica de dicha ecuación, a la que se le llama logit, consiste en convertir la probabilidad (p) en odds. De aquí surge la ecuación de la regresión logística, que es parecida a la ecuación de la regresión lineal múltiple. |
| HMM | Fácil de implementar. | • Puntuación: Dada una secuencia de observaciones y un modelo, ¿cómo calcular la probabilidad de observar la secuencia vista dado el modelo?<br>• Reconocimiento de estados: Dada una secuencia de observaciones y un modelo, ¿cuál es la secuencia de estados que mejor "explica" las observaciones?<br>• Entrenamiento: Dado un conjunto de observaciones de entrenamiento ¿Cómo se ajusta los parámetros del modelo para maximizar la probabilidad de observar el conjunto | Modelo probabilístico el cual tiene una serie de estados conocidos y otros estados que están ocultos para el observador, al no conocer ciertos estados las cadenas ocultas utilizan probabilidades basándose en hechos ya ocurridos. Esto con el objetivo de determinar los parámetros desconocidos (ocultos) de dicha cadena. |

| | | | |
|---|---|---|---|
| | | de entrenamiento dado el modelo? | |
| Montecarlo | Directo y flexible | Bastante tiempo de computo. Puede no brindar la respuesta correcta. | Es una técnica numérica para calcular probabilidades mediante la utilización de números aleatorios, esto permite calcular estadísticamente el valor final de una secuencia de sucesos no deterministas. |
| Máxima entropía | • Permite distinguir el conjunto de salida.<br>• Es útil cuando no se tiene ordenado el dataset. | Cuando hay muchas limitaciones para satisfacer, se necesitan técnicas rigurosas para encontrar la solución óptima. | Este proceso se usa para estimar cualquier distribución de probabilidad.<br>Normalmente busca la distribución de probabilidad que este menos sesgada. Cuando no se tiene conocimiento de información y se encuentra la distribución menos sesgada, esto quiere decir que contiene la mayor cantidad de información relacionada. |
| CRF | Puede acomodar más fácil la información que las cadenas ocultas de Markov(HMM). | Es complejo el reentrenamiento del modelo | Toma un conjunto de datos y a cada dato le asigna una etiqueta y utiliza la probabilidad de la secuencia observada, es representado mediante un grafo no dirigido, las aristas representan dependencia entre variables y los vértices son variables aleatorias a las cuales se les debe hallar la distribución de probabilidad. |

Fuente: Autores.

Se validan todos los algoritmos en la plataforma WEKA y se descartan el algoritmo de clasificación de Montecarlo y Conditional random field (CRF), debido a que no están implementados en la plataforma y sus principales desventajas que como por ejemplo en Montecarlo el que no pueda brindar la respuesta correcta puede influir en los resultados de manera directa en los resultados del caso de estudio y CRF con su complejidad al re-entrenar el modelo.

Para el desarrollo del proyecto se implementan los siguientes algoritmos:

- Naive Bayes.
- KNN.
- Red Bayesiana
- SVM
- Red Neuronal
- Árbol de decisión.

Se escogió el algoritmo de **Naive Bayes** porque brinda resultados confiables al realizar el entrenamiento y al ser un modelo probabilístico que construye modelos para predecir posibles resultados entrenándose con pocos datos de entrenamiento y al no ser sensitivo a características pocos relevantes, brinda resultados confiables para su análisis con respecto al caso de estudio.

Se escogió el algoritmo de **KNN** porque tiene buen desempeño en el entrenamiento de textos pese a sus desventajas principales que son: la dificultad de encontrar un k óptimo y que es sensible con las características irrelevantes, también se tiene en cuenta la ventaja propuesta y es que este algoritmo tiene facilidad en su implementación.

Dicho desempeño se muestra en el desarrollo del proyecto.

Se escogió el algoritmo de **Red Bayesiana** porque al construir grafos dirigidos acíclicos permite al algoritmo tener una buena predicción de textos, este algoritmo es fácil de entender y puede modelar sistemas complejos que para el caso de estudio serian textos complejos o con palabras poco comunes y tiene una buena predicción por que toma varios atributos para realizar una predicción.

Se escogió el algoritmo de **SVM** por su buen desempeño tanto en el entrenamiento como como en la clasificación de resultados ya que tiene una buena respuesta a problemas típicos que, para este caso de estudio, sería la conjugación de palabras en textos comunes, una característica importante es que este algoritmo se puede aplicar a una base de datos que en ella contenga conjuntos de datos complejos y que tengan ruido (Palabras no esperadas) y/o características irrelevantes.

Se escogió el algoritmo de **Red Neuronal** porque es uno de los que en la literatura muestran que tiene mejor rendimiento debido a que se adaptan a cualquier cambio que se produzca y tiene tolerancia a fallos que le permite seguir funcionando correctamente, pese a que se producto un fallo o un resultado no esperado, se escoge el algoritmo pese a su principal desventaja la cual es que tarda mucho tiempo en entrenarse.

Se escogió el algoritmo de **Árbol de decisión** porque al modelar el comportamiento o esquema de un árbol y brindarles a las hojas etiquetas de clasificación, permite que todas las opciones de un problema o escenario sean analizadas y brinde un resultado que es el porcentaje de clasificación de un comunicado.

No se toman en cuenta algoritmos basados en la **regresión logística** debido a que este algoritmo tiene un buen desempeño cuando se tienen clases dicotómicas (SI o NO) y para este caso de estudio se tienen tres clases que son pacifista, guerrerista o neutra, por esta razón utilizar este algoritmo produce que no se obtengan los resultados esperados.

Tampoco se toma en cuenta el algoritmo basado en **HMM** (Hidden Markov Model o cadenas ocultas de Markov) por las múltiples desventajas encontradas y que es un modelo que se utiliza principalmente para predecir estados desconocidos por medio de una serie de estados vistos o conocidos y otros ocultos, para este caso de estudio no tenemos los estados ocultos por lo cual el utilizarlo no es útil

No se utiliza el algoritmo de **Máxima Entropía** por su principal desventaja, Cuando hay muchas limitaciones para satisfacer, se necesitan técnicas rigurosas para encontrar la solución óptima, en el desarrollo del proyecto ocasiona el no obtener los resultados esperados, de igual manera en la plataforma escogida para el desarrollo del proyecto, no se tiene el algoritmo sino una aproximación del este lo cual reduce los resultados esperados al utilizar este algoritmo.

## 4. HISTORIAS DE USUARIO

‘

En el desarrollo de las historias de usuarios se desarrollan en orden ascendente en donde tendrán una misma prioridad, salvo que no se puede realizar una historia de usuario sin terminar la inmediatamente anterior. La duración de cada una de las historias de usuario no debe exceder una semana (siete días), en caso tal, de que se termine una historia de usuario antes del límite máximo se podrá comenzar con la siguiente añadiéndole los días faltantes de la historia de usuario inmediatamente anterior, por ejemplo, si la HI – 01 se terminó en cinco días el tiempo estimado para el desarrollo de la HI – 02 será de nueve días.

Tabla 6. Lista de requerimientos.

| | Enunciado de la historia | | | | Criterios de aceptación | | | |
|---|---|---|---|---|---|---|---|---|
| Identificador (ID) de la historia | Rol | Característica / Funcionalidad | Razón / Resultado | Número (#) de escenario | Criterio de aceptación (Título) | Contexto | Evento | Resultado / Comportamiento esperado |
| HI - 01 | Encargado de seguimiento | Necesito que los encargados del seguimiento validen los entornos y plataformas, relacionados con la implementación de procesos de clasificación de textos cortos. | Con la finalidad de que sea escogido un entorno o plataforma. | 1 | Ventajas y desventajas de utilizar entornos y plataformas | Se debe conocer las características generales del entorno y la plataforma, que permita identificar las ventajas y desventajas que poseen al ser utilizada. | Al estudiar los entornos y las plataformas | Se conocerá el funcionamiento de los entornos y las plataformas, también debe existir una tabla que contenga ventajas y desventajas de utilizar dichos entornos y plataformas. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| HI - 02 | Encargado de seguimiento | Necesito que los encargados del seguimiento validen los algoritmos más usados para la clasificación de textos cortos. | Con la finalidad de que sea escogido un grupo de algoritmos para implementarlos en el desarrollo del caso de estudio. | 1 | Ventajas y desventajas de utilizar los algoritmos más usados para la clasificación de textos cortos. | Se debe construir una tabla con las características generales del algoritmo en la cual se detallen las ventajas y desventajas que posee. | Al estudiar los algoritmos | Se conocerá el funcionamiento de los algoritmos, también debe existir una tabla que contenga ventajas y desventajas de utilizar dichos algoritmos. |
| | | | | | | | | |
| HI - 03 | Desarrollador | Necesito que el desarrollador conozca la plataforma. | Con la finalidad de no entorpecer procesos futuros. | 1 | Dudas sobre el manejo de la plataforma | Al no tener claro el proceso que realiza un módulo, parte y/o acción de la plataforma. | El desarrollador debe instruirse con las TIC's | Para resolver sus dudas de manera autónoma, y así, al final del proceso debe conocer cómo funciona la plataforma para el desarrollo del caso de estudio. |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HI - 04 | Administrador de Base de datos | Se debe validar la base de datos suministrada. | Esta no debe presentar inconvenientes por datos incompatibles. | 1 | Adaptar la base de datos | En caso de que la base de datos suministrada por la universidad no sea entregada en un formato que maneje la plataforma | Cuando se esté entrenando la maquina | El administrador de base de datos deberá cambiar el formato y/o adaptar los datos para que la plataforma lea el dataset. |
| | | | | | | | | |
| HI - 05 | Entrenador | Necesito que el entrenador valide el dataset entregado y lo ingrese para el entrenamiento de los algoritmos en la plataforma. | Con la finalidad de entrenar los algoritmos. | 1 | Ingresar la base de datos | La base de datos deberá ser ingresada en la plataforma weka en formato .arff este se separa en tres secciones: título o identificador, la cabecera y los datos, este formato rige las siguientes especificaciones:<br>• En el identificador sigue la siguiente norma: comienza con un @RELATION seguido del nombre del set de datos.<br>• Las declaraciones de atributos, usaremos @ al comienzo seguido del identificador ATTRIBUTE, luego el nombre del atributo y finalmente el tipo de dato.<br>• La sección de los datos debe comenzar con la declaración @DATA seguido de un interlineado, después contendrá en cada línea los datos aclarando que cada dato perteneciente a una característica de un objeto | Al ingresar los datos en la plataforma | La plataforma quedara lista para que el desarrollador implemente los algoritmos y se aplique el concepto de Machine Learning. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | deberá estar separado con una coma, para finalizar la sección se denota un %. | | |
| | | | | | | | | |
| HI - 06 | Desarrollador | Necesito que el desarrollador implemente los algoritmos seleccionados en la plataforma. | Con la finalidad de obtener los resultados de clasificar el algoritmo. | 1 | Configurar los algoritmos. | El desarrollador configurara los algoritmos en la plataforma. | Al implementar el algoritmo en la plataforma. | La plataforma mostrara la configuración realizada. |
| | | | | 2 | Validar los resultados del entrenamiento | Se validara los resultados de cada algoritmo. | Al poner en marcha la clasificación. | Se validara que el algoritmo entrene correctamente y muestre una tabla de resultados |
| | | | | | | | | |
| HI - 07 | Desarrollador | Necesito que genere la vista del programa utilizando Java, importando y utilizando las librerías de WEKA. | Para que el programa sea fácil de utilizar por el usuario final. | 1 | Crear interfaz gráfica. | Todo el proyecto se debe empaquetar desde Java con sus respectivas librerías. | Ejecución del programa | Se mostrara el proyecto de estudio, en un programa ejecutable para sistemas operativos Windows. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HI - 08 | Encargado pruebas. | Necesito que valide los resultados obtenidos. | Con la finalidad de que se tenga certeza que los resultados son correctos. | 1 | Validar los resultados del entrenamiento. | El encargado de realizar las pruebas validara los resultados de los algoritmos e informara que algoritmo tiene un mejor resultado en la clasificación. | Validar los resultados. | Se informa cual es el mejor algoritmo para la clasificación de texto corto teniendo en cuenta el caso de estudio. |
| | | | | | | | | |

Fuente: Autores.

# 5. DESARROLLO DEL COMPONENTE

## 5.1. WEKA

Se ha mencionado que WEKA ha sido desarrollada en el lenguaje de programación Java y de su fácil integración con el lenguaje de programación Java, mediante una librería (weka.jar) la cual puede ser descargada gratuitamente del sitio oficial de WEKA o también, cuando el usuario descarga el programa, puede hallar dicha librería en el directorio donde fue instalado el programa, en este directorio también se puede encontrar toda la documentación referente a las diferentes modalidades que WEKA brinda para la implementación de los diferentes algoritmos, aparte se encuentra la documentación que permite incluir WEKA en Java, se puede observar que la documentación es muy buena porque esta detallada la funcionalidad de cada algoritmo, también muestra métodos para poder implementarlos, evaluarlos, incluirle filtros, entre otras funcionalidades.

WEKA cuenta con foros en donde los usuarios realizan preguntas y varios desarrolladores de la plataforma contestan, permitiendo conocer las funcionalidades de la plataforma en detalle, dichos foros están funcionando desde hace varios años atrás por lo que permite encontrar respuestas de una manera óptima, por lo cual utilizarla tiene efectos positivos en el desarrollo del software deseado debido a que se puede invertir más tiempo en el desarrollo y no perder tiempo en investigación de preguntas ya resueltas en los mencionados foros, al analizar las características del proyecto, el objetivo principal del mismo y tener en cuenta el calendario establecido para la realización de las diferentes historias de usuario, reiteramos que WEKA es la plataforma indicada a utilizar para el desarrollo.

### 5.1.1. Estructura del conjunto de datos.

En la sección 2 se habló acerca de las plataformas y lenguajes más utilizados en la actualidad para realizar la clasificación y predicción de tweets, en dicha sección se argumentó porque se utilizó WEKA en el presente proyecto, de allí es importante recordar que la plataforma está desarrollada en el lenguaje de programación Java, la universidad de Waikato desarrolló una librería (weka.jar) lo que permite al usuario trabajar directamente en la plataforma desarrollada por ellos o permite poder incluir dicha librería en Java para que cada usuario pueda utilizar las herramientas (Algoritmos, Filtros, Evaluación, entre otros.) en el código fuente del software que se esté desarrollando.

Para mostrar cómo se desarrolló el componente de análisis, clasificación y predicción de tweets primero hay que tener claros ciertos conceptos de como maneja la información WEKA.

Ilustración 3. Estructura. arff

```
@RELATION tweets

@ATTRIBUTE comunicado STRING
@ATTRIBUTE orientacion     {pacifista,guerrerista,neutra}

@DATA
"#ProcesoDePaz El conflicto colombiano dejó 16.879 niños
víctimas por reclutamiento forzado",neutra
"Expresidente @JuanManSantos meses atrás le dije varias veces
gracias por el #ProcesoDePaz y por devolver la confianza al país
y a los jóvenes, por su inversión en educación. Hoy ratifico mi
apoyo, mi agradecimiento y mi admiración por tanto que hizo x el
país. #ColombiaLoNecesita",pacifista
"#ProcesoDePaz Los criterios jurisprudenciales que recibe la JEP
de la Corte Suprema de Justicia",neutra
"Que dejen las armas de 50 años a cambio de nada?
Intetesante.... @IvanDuque #Eln #procesoDePaz",guerrerista
"#ProcesoDePaz Ley de amnistía condicionada a la satisfacción de
los derechos de las víctimas",guerrerista
"#DebateW | Iván Duque recibe un país resquebrajado,
desinstitucionalizado por lo que sucedió en el acuerdo con las
Farc: Hernán Prada con #VickyDávilaEnLaW",guerrerista
"No inventé! No sea oportunista, no politice ni engañe
@ALVAROHPRADA no intente hacer del nuevo régimen (gobierno) un
oasis, no menosprecie ni haga trizas con sus palabras un
#ProcesoDePaz un #AcuerdoDePaz, no reduzca procesos políticos
sólo por un partido",pacifista
```

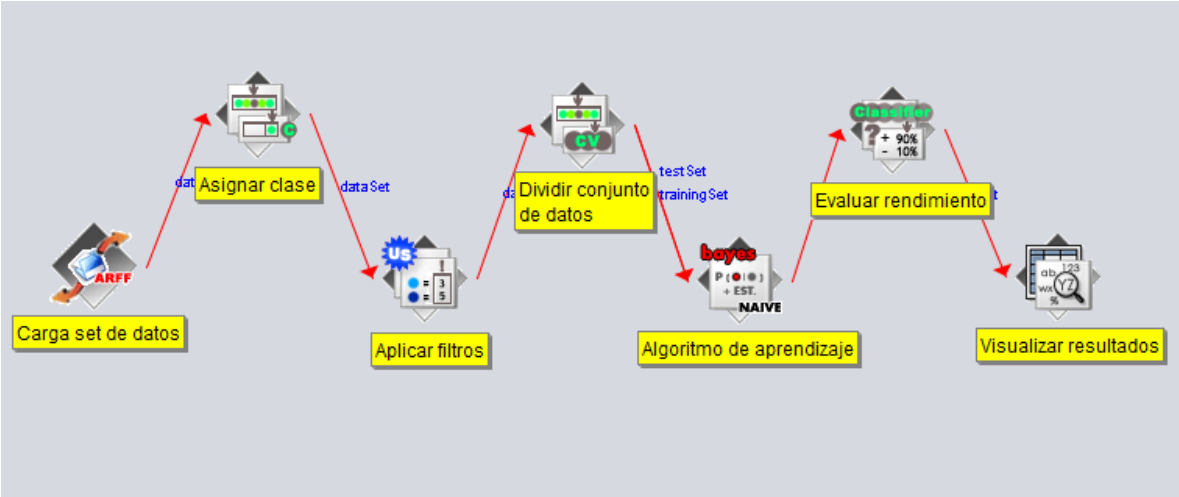
Fuente: Autores.

La ilustración 3, se muestra como debe ser la estructura de los tweets para poderlos analizar, a pesar de que WEKA maneja archivos CSV, XLSS, entre otros. Se decidió optar por esta estructura debido a que en la revisión de proyectos similares los usuarios manifestaban inconvenientes en la carga de datos de archivos con extensión distinta a .arff, en la ilustración se observa como la estructura se divide en 3 partes, la primera parte es la etiqueta @RELATION, esta etiqueta permite asignarle un nombre a la base de datos, para el caso del proyecto se le asigna como nombre “tweets”, la segunda parte empieza por la etiqueta @ATRIBUTTE, allí permite asignar la cantidad de atributos con las que cuenta el set de datos, observando la ilustración se puede evidenciar que se cuenta con 2 atributos, el primero va a ser cada tweet y al ser de tipo texto se declara como STRING, acá es importante resaltar que para los elementos de tipo STRING que contengan espacios deben ser colocados entre comillas dobles, estos archivos manejan un estándar y es que las clases a las que se desea clasificar deben

declararse como el ultimo atributo del archivo, WEKA y cualquier plataforma que maneje este tipo de archivos por defecto va a decir que este atributo va a ser el que contiene la clase y a pesar de que este se puede cambiar es recomendable seguir un estándar, la tercera parte consiste en los datos como tal y es denotado por la etiqueta @DATA, el único requerimiento en esta sección es que los datos sean separados por comas y estén en el orden en el que se declararon los atributos.

### 5.1.2. Flujo de datos en weka.

Ilustración 4. Flujo de datos en WEKA.



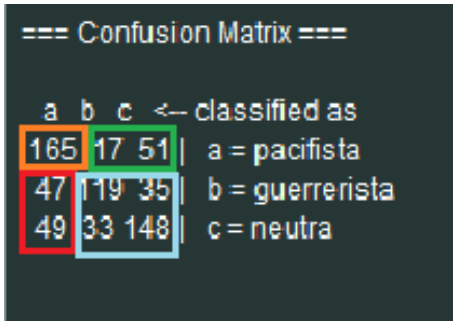

Fuente: Autores.

Después de tener claro la estructura de un set de datos en WEKA se debe conocer el flujo de los datos a lo largo de la ejecución del programa, en la ilustración 4 se ve cómo se mueven los datos en el tiempo, primero se carga el archivo en formato arff, luego se le debe asignar una clase, en este punto lo que la plataforma hace en esa instancia es identificar los datos y le va a decir cuál va a ser la clase a identificar, después se puede agregar filtros dependiendo del conjunto de datos que se tenga, si los datos son de tipo numérico no es necesario aplicar un filtro, pero si los datos son de tipo texto es obligatorio agregar un filtro para convertirlos en tipo numérico, en esta instancia WEKA permite agregar técnicas de pre-procesamiento de textos como lo son palabras de paro, stemming, entre otras. En el siguiente momento se debe dividir el conjunto de datos, acá se debe tener claro que siempre se debe dejar una parte del conjunto de datos para probar el algoritmo, por lo que allí se divide el conjunto de datos en 2 sub conjuntos, el conjunto de entrenamiento y el conjunto de pruebas, también es importante recordar que ni el conjunto de entrenamiento ni el de pruebas debe ser tan pequeño por esto se debe encontrar un balance entre los 2 subconjuntos, para

esto existe la validación cruzada la cual divide el conjunto de datos en un numero de partes iguales, es importante evaluar nuestro conjunto de entrenamiento y hallar el mejor número de partes para dividir el set, esto con la final de pasarle los subconjuntos de datos al algoritmo para que este pueda proceder a entrenarse y luego a probar con el conjunto de pruebas, el penúltimo paso es construir un evaluador el cual permitirá, mediante datos estadísticos, analizar que tan bien quedo entrenado el algoritmo, eficiencia, entre otros muchos datos. Finalmente se debe crear un visualizador para poder observar estos resultados.

### 5.1.3. Interpretar los valores de WEKA.

Ilustración 5. Matriz de confusión.




Fuente: Autores.

La matriz de confusión cuenta con 4 partes que se identifican en la ilustración 5, para realizar una explicación mejor se realizará un ejemplo de que significan los valores de la clase pacifista, el valor encerrado en el color naranja representa los TP (Verdaderos Positivos) que son los valores etiquetados como pacifistas y se clasificaron como pacifistas, los valores encerrados en color rojo representan los FN (Falsos negativos) que son aquellas instancias o comunicados que son pacifistas y el sistema las clasifica en otras clases, los valores encerrados en color verde representan los FP (Falsos Positivos) que son los valores predichos como pacifistas que deberían haber sido predichos en otra clase, los valores encerrados en el cuadro azul son los TN (Verdaderos Negativos) los cuales representan los valores clasificados en otras clases aparte de la pacifista que no deberían ser pacifistas, una matriz de confusión ideal es la que contiene en su diagonal el total de cada una de las clases y los valores que están fuera de la diagonal deben estar en cero.

De igual manera se aclara que en las columnas están las instancias clasificadas por el algoritmo y en las filas están las instancias o comunicados predichos para cada una de las clases

Ilustración 6. Matriz de resumen.


Fuente: Autores.

En la Ilustración 6, se observa una matriz de valores, los 2 primeros valores corresponden a las instancias de prueba correcta e incorrectamente clasificadas, allí se observa cuantas instancias fueron clasificadas en dicha categoría y el porcentaje que corresponde, para hallar este porcentaje simplemente se divide el número de instancias de la clase sobre el total de instancias clasificadas en el ejemplo las instancias correctamente clasificadas son:

$$\frac{432}{664} \times 100 = 65.0602\%$$

De igual manera se realiza el proceso para las incorrectas:

$$\frac{232}{664} \times 100 = 34.9398\%$$

El valor de la estadística **Kappa** representa el grado de acuerdo entre las clasificaciones y las clases verdaderas (concordancia), para hallar el valor de Kappa se deben hallar 2 valores, **precisión observada** y **precisión esperada**

La **precisión observada** se halla de la siguiente manera:

$$precision\ observada = \frac{165 + 119 + 148}{644} = 0.6506$$

Se suman los valores de tp para cada una de las clases y se divide en el total de comunicados.

Luego se debe hallar la **precisión esperada** se realiza de la siguiente manera:

$$233 * \frac{261}{664} = 91.5858$$

Se toma el número total de instancias de una clase y se multiplica por el total de las instancias clasificadas en una clase sobre el número total de comunicados.

$$201 * \frac{169}{664} = 51.1581$$

$$230 * \frac{234}{664} = 81.0542$$

Posteriormente se suman los valores obtenidos y se divide sobre el total de instancias o comunicados cortos.

$$precision\ esperada = \frac{91.5858 + 51.1581 + 81.0542}{664} = 0.3370$$

Luego aplicamos la formula y se halla el valor de Kappa:

$$Kappa = \frac{precisión\ observada - precisión\ esperada}{1 - precisión\ esperada}$$

Al remplazar

$$Kappa = \frac{0.6506 - 0.3370}{1 - 0.3370} = 0.4730$$

Kappa es una medida de qué tan cerca las instancias clasificadas por el clasificador de aprendizaje automático **coincidieron** con los datos etiquetados como **verdaderos**, brinda un mejor indicador de cómo se desempeñó el clasificador en todas las instancias siguiendo, un estándar, si kappa está en: > 0.75 es excelente, 0.40 - 0.75 se toma como justo a bueno, y <0.40 como pobre o no clasifico correctamente (Bakeman, McArthur, Quera, & Robinson, 2014).

Para el ejemplo, el resultado de Kappa nos brindó un resultado de 0.473, lo cual quiere decir que tiene una buena clasificación de instancias verdaderas con respecto a la base de datos.

Para hallar el error absoluto **MAE** (Mean Absolute Error o Error absoluto medio), se tiene que aplicar la siguiente formula, donde:

$x_i$ es: precisión esperada
$y_i$ es: precisión observada
$N$ Es: el total de los datos o el total de un conjunto de predicciones o comunicados.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |x_i - y_i| = 0.2435$$

MAE es utilizado para medir la magnitud promedio de los errores de un conjunto de comunicados, sin considerar su dirección o su signo. El MAE es una puntuación lineal, lo que significa que todas las diferencias individuales se ponderan por igual en el promedio.

El siguiente valor que se observa en la Ilustración 10 es la raíz del error cuadrático medio (Root Mean Squared Error (**RMSE**)) el cual se calcula con la siguiente ecuación:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - y_i)^2} = 0.4148$$

RMSE es una regla de puntuación cuadrática que se utiliza para medir la magnitud promedio del error y al explicar la fórmula: se le aplica raíz cuadrada al promedio de las diferencias cuadradas entre la precisión esperada y la precisión observada todos esto sobre la muestra.

MAE y RMSE se pueden usar al mismo tiempo para diagnosticar la variación en los errores en un conjunto de predicciones.

El RMSE siempre será mayor o igual al MAE; Cuanto mayor sea la diferencia entre ellos, mayor será la varianza (dispersión) en los errores individuales en la muestra. Si el RMSE = MAE, entonces todos los errores son de la misma magnitud

El error absoluto relativo **RAE** (Relative Absolute Error) es hallado mediante la siguiente formula:

$$RAE = \frac{\frac{1}{N}\sum_{i=1}^{N} |X - \theta_i|}{\frac{1}{N}\sum_{i=1}^{N} |\alpha - \theta_i|} = 0.5490$$

Este error indica la calidad de la medición, entre más bajo sea el valor quiere decir que la medición es mejor, alfa representa la media de los datos.

La raíz del error cuadrático relativo RRSE, se calcula con la siguiente formula:

$$RRSE = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}(X-\theta_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(\alpha-\theta_i)^2}} = 0.8808$$

RRSE es la raíz cuadrada de la media de errores al cuadrado y es una medida de precisión, su resultado nunca tiene signo negativo y un valor similar a 0, informa un ajuste perfecto de los datos.

Ilustración 7. Matriz de precisión detallada por clase

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0,708 | 0,223 | 0,632 | 0,708 | 0,668 | 0,474 | 0,827 | 0,738 | pacifista |
| | 0,592 | 0,108 | 0,704 | 0,592 | 0,643 | 0,511 | 0,856 | 0,757 | guerrerista |
| | 0,643 | 0,198 | 0,632 | 0,643 | 0,638 | 0,444 | 0,820 | 0,697 | neutra |
| Weighted Avg. | 0,651 | 0,179 | 0,654 | 0,651 | 0,650 | 0,475 | 0,834 | 0,730 | |

Fuente: Autores.

En la Ilustración 7. se observa el valor de **precision**, este es calculado mediante la siguiente formula:

$$Precision = \frac{TP}{TP + FP}$$

Este valor es dado por cada clase y lo que indica es el porcentaje de cuantos valores predichos en una clase son verdaderamente de esa clase.

El valor **recall** o también llamado sensibilidad es calculado con la siguiente formula:

$$Recall = \frac{TP}{TP + FN}$$

Al igual que la precisión este valor es referenciado en cada clase e indica de cuantos valores de la clase que debían ser seleccionados fueron seleccionados.

La **F-Measure** es calculada de la siguiente manera:

$$F - Measure = 2 * \frac{Precisión * Recall}{Precisión + Recall}$$

Esta medida representa un balance entre la precisión y el recall y representa que tan bien predice el algoritmo.

El valor **MCC** se es calculado:

$$MCC = \frac{(TP * TN) + (FP * FN)}{\sqrt{(FP + TP)(TP + FN)(TN + FP)(TN + FN)}}$$

Este valor indica la calidad de la clasificación de cada clase.

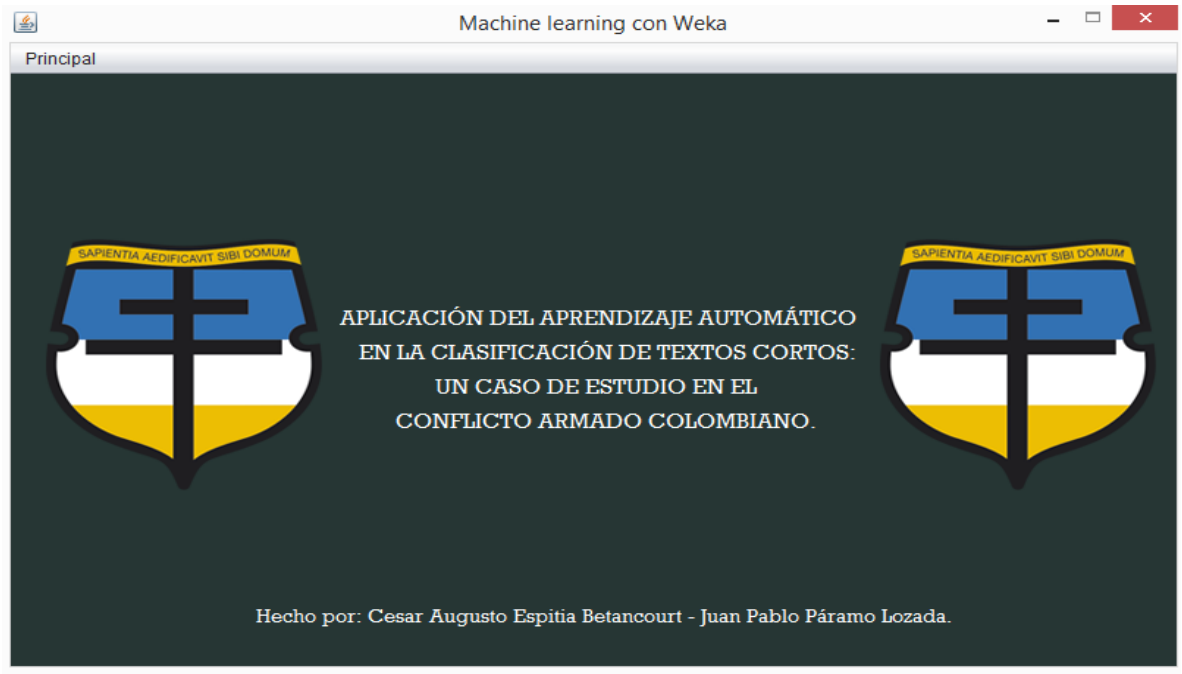
La curva **ROC** da una idea al usuario de cómo es el rendimiento general del algoritmo, para realizar esta grafica se utiliza los TP vs FP, el valor que indica en la tabla entre más cercano a 1 quiere decir que el rendimiento general del algoritmo es bueno.

El área **PRC** se enfoca en cada clase, este valor dice como es el comportamiento de cada clase para esto utiliza precisión vs recall, de igual manera que la curva ROC entre más cercano a 1 mejor el comportamiento.

### 5.1.4. Desarrollo en Java.

Java es un lenguaje de programación con gran acogida y amplia documentación, también es un lenguaje conocido a nivel mundial que permite integrar otros tipos de software en él. Al estudiar WEKA se puede concluir que Java es el lenguaje de programación indicado para el desarrollo del presente proyecto debido a que en la metodología se planteó un desarrollo ágil y la fácil integración de la plataforma con el lenguaje agiliza el proceso de desarrollo siendo una correcta aplicación de la metodología planteada.

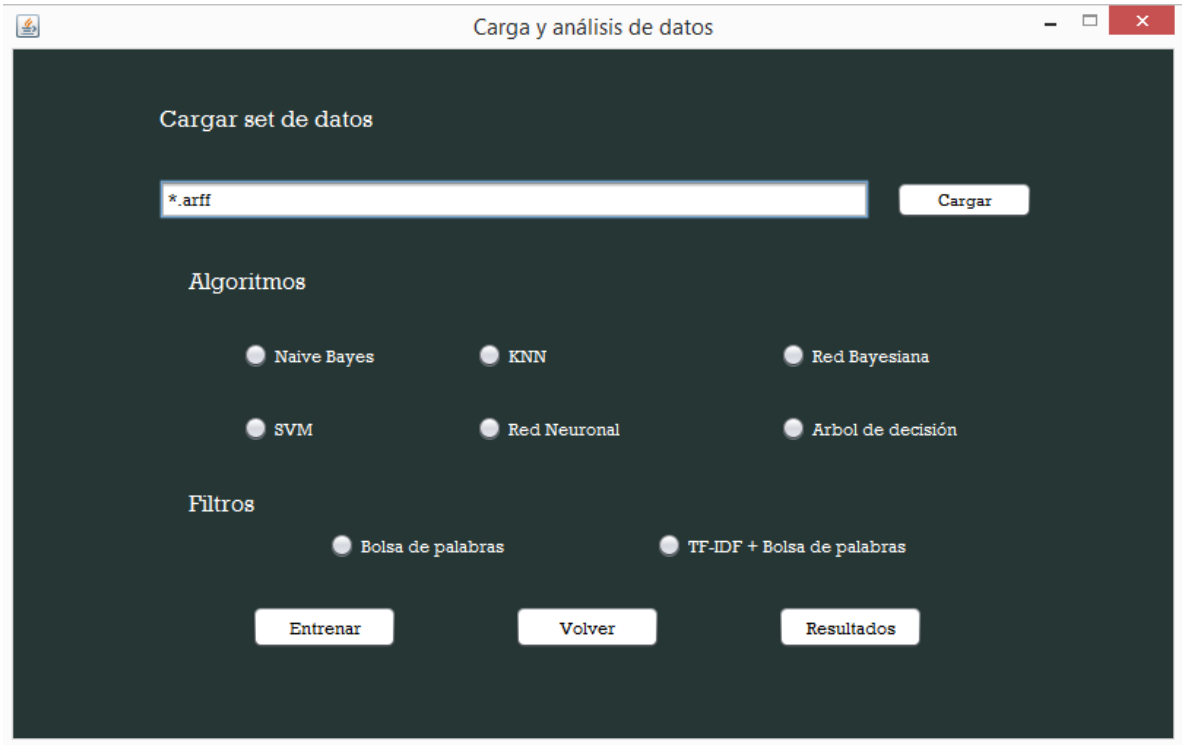
Ilustración 8. Página de presentación.




Fuente: Autores.

Se puede observar en la ilustración 8, la página de presentación del software desarrollado en el proyecto, esta página cuenta con el logo de la universidad, el título del presente proyecto y los nombres de los autores, adicional a esto en la parte superior se observa un menú con una pestaña llamada principal que permite al usuario comenzar a interactuar con el software, al darle click en principal, se despliega un submenú llamado “Carga y análisis” el cual permite al usuario ir directamente a realizar el proceso de entrenamiento y análisis de resultados.

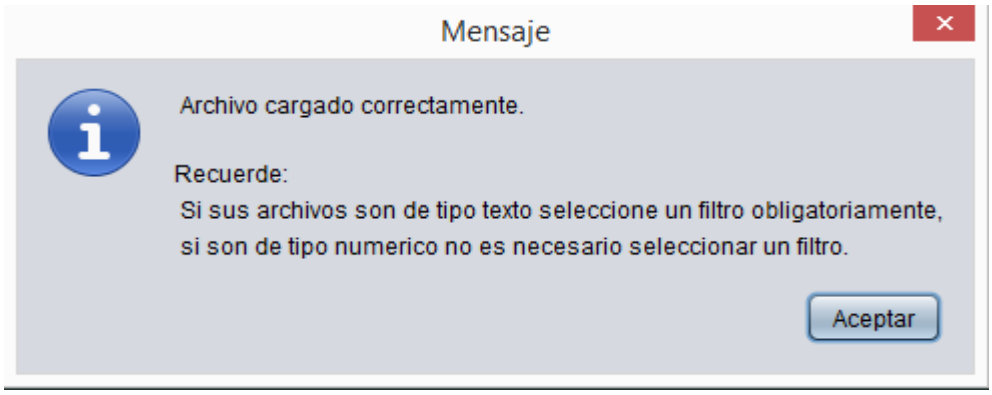
Ilustración 9. Carga y análisis de datos.




Fuente: Autores.

La segunda pantalla se observa en la Ilustración 9, esta cuenta con opciones sencillas y entendibles para el usuario, en la sección de “Cargar set de datos” se muestra el botón “Cargar” el cual al darle click permite al usuario explorar en su equipo y poder adjuntar la base de datos a procesar, cuando este cargado el archivo automáticamente el programa le genera un mensaje al usuario como se ve en la Ilustración 10.

Ilustración 10. Mensaje de carga de datos.

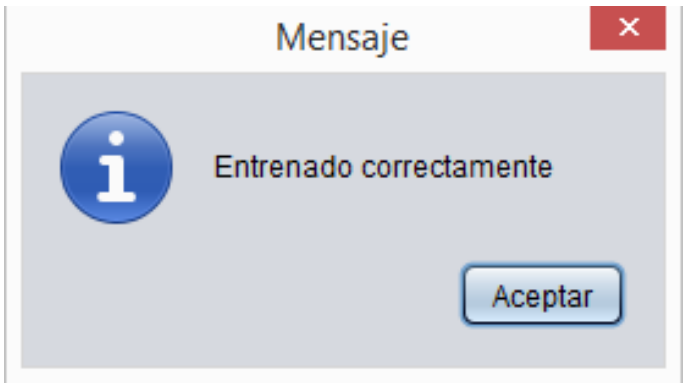



Fuente: Autores.

De igual manera se verifica si se ha cargado correctamente el archivo o no y además muestra un mensaje para la correcta utilización del software.

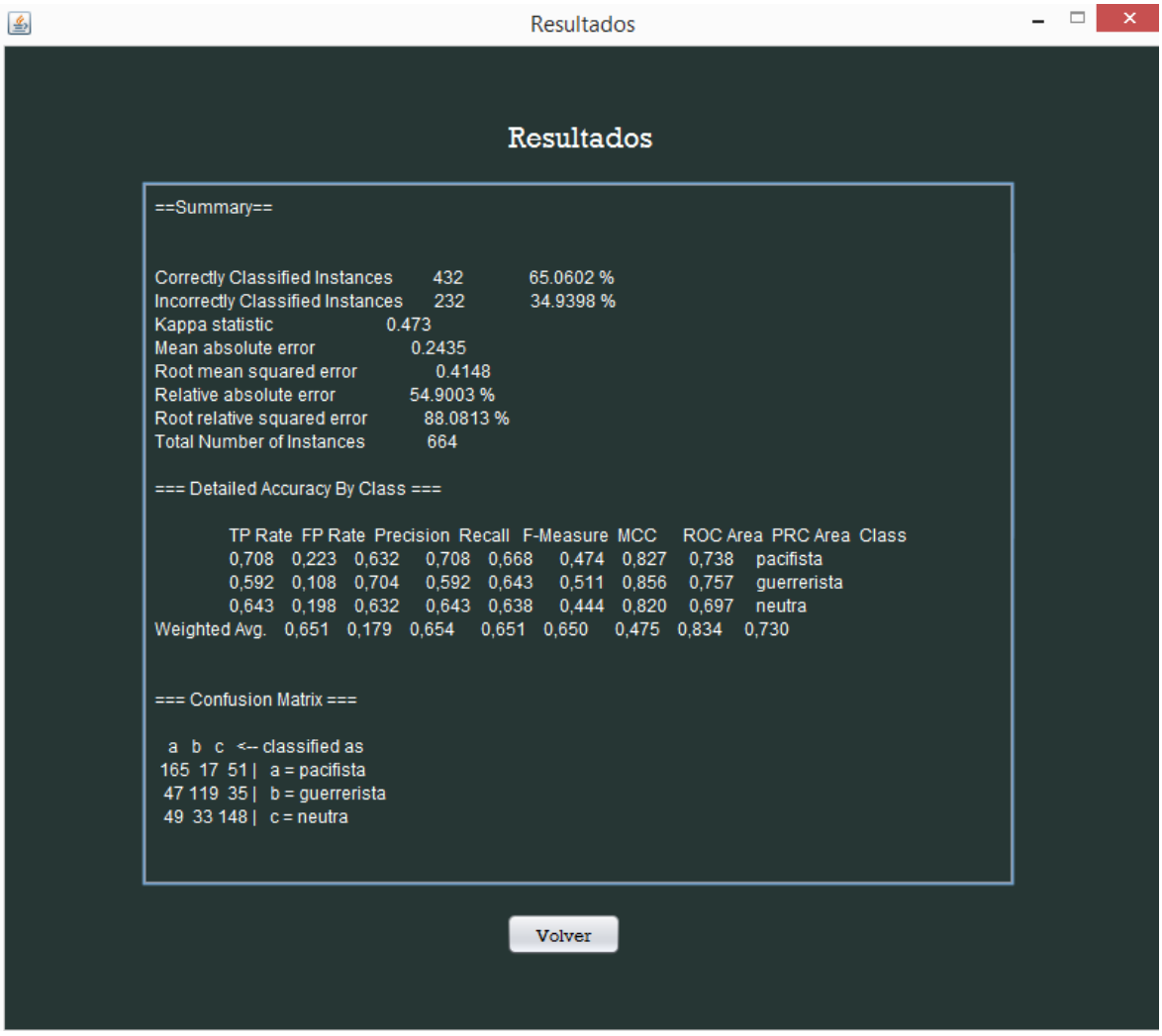
En la segunda sección “Algoritmos” de la ilustración 9, el software permite al usuario seleccionar uno de los seis algoritmos, esto con el fin de entrenarlo; posteriormente el usuario puede o no seleccionar un filtro para sus datos, si sus datos son de tipo numérico no es necesario seleccionar el filtro pero si son de tipo texto es necesario que seleccione un filtro, luego de esto se le da click en el botón entrenar para que el software entrene el algoritmo seleccionado, al haber entrenado se mostrará un mensaje de si se ha entrenado correctamente o no, el mensaje se puede observar en la Ilustración 11.

Ilustración 11. Mensaje de entrenamiento correcto.




Fuente: Autores.

Ilustración 12. Tablas de resultados.




Fuente: Autores.

En la Ilustración 12, se observan los resultados que muestra el programa, están divididos en 3 secciones, la primera es la sección de resumen, la segunda se encuentra la precisión detallada por cada clase y la tercera es la matriz de confusión, estos resultados son analizados en la sección posterior del presente proyecto llamada “Resultados”, el botón “Volver” permite al usuario regresar a la sección “Carga y análisis” para que pueda realizar el mismo procesamiento de entrenamiento con diferentes algoritmos y filtros.

# 6. RESULTADOS

## 6.1. ADICIÓN DE TWEETS

Como se ha mencionado a lo largo del proyecto, la facultad de psicología entregó una Base de datos la cual consistía en 317 tweets que están divididos de la siguiente manera: 92 de los tweets corresponden a comunicados cortos con orientación guerrerista, 108 con orientación neutra y 117 pacifista.

Al realizar el entrenamiento con el algoritmo Naive Bayes y el filtro Bolsa de Palabras el software brindó los siguientes resultados.

Tabla 7. Resultados con la base de datos suministrada.

| Filtro \ Algoritmo | Naive Bayes | KNN | Red Bayesiana | SVM | Árbol de decisión |
|---|---|---|---|---|---|
| Bosa de Palabras | 51,4196 | 36,9085 | 36,9085 | 49,5268 | 41,0095 |
| B.P. + TF – IDF | 43,8486 | 37,5394 | 37,5394 | 47,0032 | 40,0631 |

Fuente: Autores.

Lo cual no es el resultado esperado para el proyecto porque las instancias clasificadas correctamente no sobre pasan el 51.4196% del total de las instancias que están en la base de datos, esto conlleva a que no se tiene un buen entrenamiento del algoritmo, sin embargo, se pudo evidenciar que el algoritmo con la mejor clasificación fue el Naive Bayes que utilizo el filtro de Bolsa de Palabras con el cual obtuvo un resultado de instancias correctamente clasificadas: 51.4196%, que sigue siendo un valor muy bajo y no garantiza que el algoritmo haya sido entrenado correctamente.

Por esta razón se realizó ajustes en la base de datos entregada, dichos ajustes consisten en:

- Corregir errores ortográficos.

Se corrigen los errores ortográficos ya que hay algunos escritores de estos comunicados que siguen las reglas de la gramática y escriben correctamente, esta razón ocasiona que para los escritores que no siguen las reglas de la gramática y escriben palabras como: q en vez de que o + en vez de más entre otras o también que escriban palabras con errores ortográficos, esto tiene repercusiones negativas para el entrenamiento del algoritmo.

- Eliminar palabras que pueden confundir el algoritmo

Algunos de los autores de los comunicados cortos se expresan con palabras que en ocasiones solo entienden pocas personas y el uso de estas palabras no se realiza con frecuencia por lo que su poca aparición en la base de datos puede ocasionar que el algoritmo no entrene una orientación adecuada para otros comunicados, algunas de estas palabras son: Abigeato, Caución.

- Cambio de palabras

Es normal que en una red social se expresen con Libre albedrío, sin embargo, para el caso de estudio este hecho tiene repercusiones negativas para el entrenamiento de cualquier algoritmo, debido a que muchos de los comunicados cortos tienen uso excesivo de los Hashtag, como, por ejemplo

*"#Encuestas...Caricatura de @matadoreltiempo #ConLaOrejaRoja #Opinión #Crítica #ProcesoDePaz #PazenColombia @JuanManSantos sigue con el #procesodepaz pese a su #desaprobación"*

Por este motivo se realizó cambio de palabras para mejorar el entrenamiento del algoritmo teniendo en cuenta que algunos Hashtag no aparecen con mucha frecuencia, por ejemplo:

El *#Encuestas* se cambió a *Las encuestas*, con esto garantizamos un mejor entrenamiento para cualquier algoritmo.

- Adición de Tweets.

Con los resultados obtenidos en el entrenamiento de la base de datos entregada, se observa que la cantidad de tweets es muy baja por lo cual se tiene que agregar más tweets a la base de datos, varios autores sugieren que la cantidad de instancias en una base de datos para aplicar Machine Learning debe ser alta y sugieren que sea superior a 500.

Con el fin de no entorpecer la clasificación brindada por la facultad de psicología, se analizaron los tweets entregados y se buscaron en la red social Twitter, comunicados similares emitidos por los actores del conflicto mencionados en la base de datos y se le brindó una orientación con lo estudiado en el análisis hecho por la facultad de psicología, se agregaron tweets como:

*"Presidente asegura que el único camino para la paz es el perdón y la reconciliación"* para el cual se le brindo una orientación pacifista

- Eliminación de Tweets.

Al realizar la verificación de los tweets entregados da como resultado que algunos tweets tienen que ser eliminados de dicha base de datos, uno de los motivos es que un comunicado equis puede tener dos orientaciones como en el siguiente ejemplo que puede tener una orientación pacifista y guerrerista al mismo tiempo, la consecuencia principal es que confunda al algoritmo porque, al tener dos orientaciones, las palabras que se usaron para entrenar una de ellas y que este en otra orientación tiende a confundir el algoritmo y más específicamente la orientación.

*Invitamos a todos los colombianos que se pronunciaron a favor de la paz a defender lo pactado, a acompañar a la #BancadaPorLaPaz en su trabajo por impedir que se vuelva trizas el acuerdo de paz, con la excusa de que es mejorable. @FARC_EPueblo @TimoFARC*

El comunicado mencionado anteriormente la facultad de psicología le brindo una orientación pacifista y se concuerda con esa decisión, sin embargo, este comunicado tiene adicional una orientación guerrerista debido que en él están incluidas palabras que se utilizan para este tipo de comunicados, por esta razón se eliminó el tweet de la base de datos con el fin de no confundir los algoritmos y garantizar una correcta clasificación de los mismos.

Otro escenario, es que se le brinde una orientación a un comunicado corto que no tiene relación alguna con el caso de estudio, que es el conflicto armado colombiano, comunicados cortos como, por ejemplo:

*"Shakira le mete un gol a la pobreza y así ayuda con la construcción de un mejor, ella construirá un colegio en Barranquilla"*

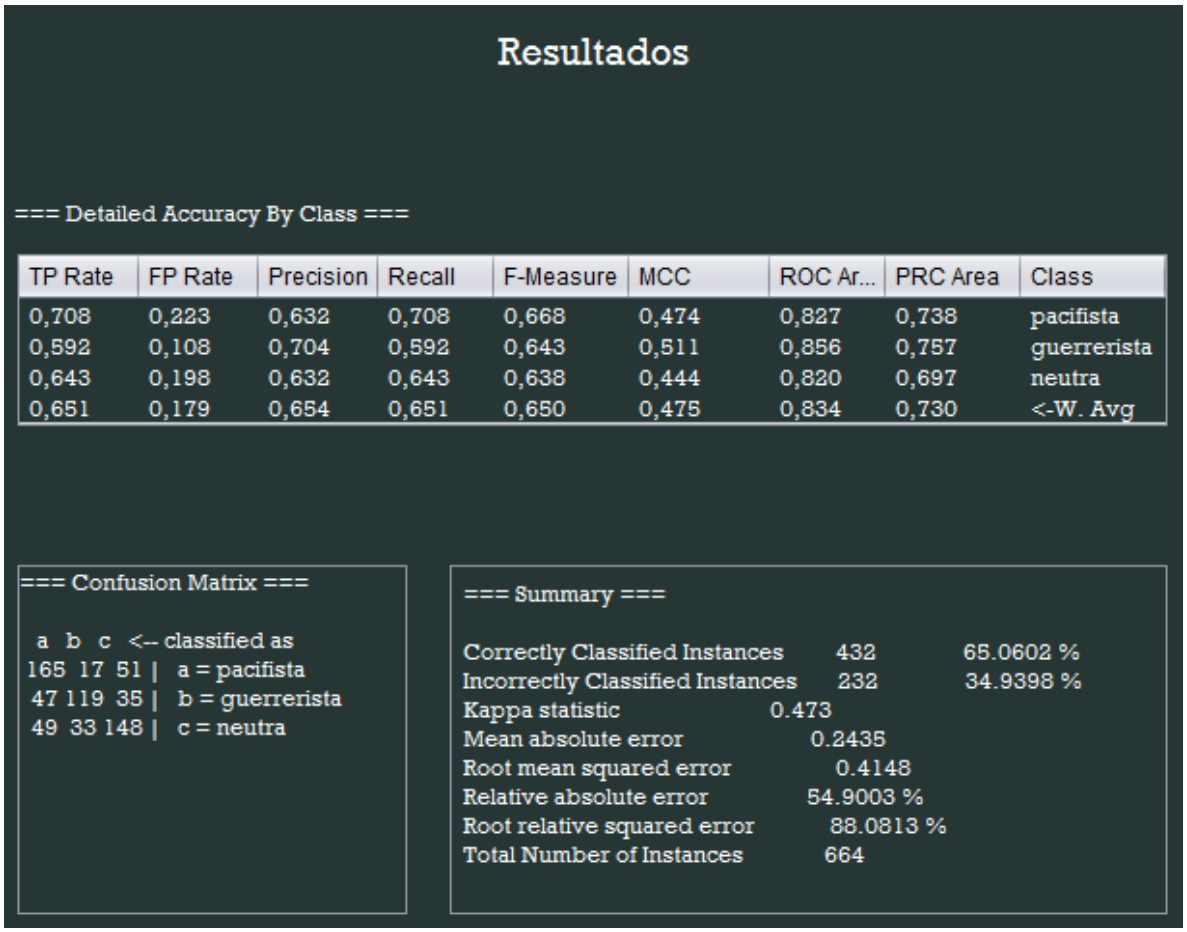
Que, aunque tenga una orientación pacifista no está relacionado con el conflicto armado colombiano y es que este acto está más relacionado a una obra de caridad realizada por la artista colombiana.

Con los cambios antes mencionados sobre la base de datos logramos un mejor entrenamiento que se mostraran en la siguiente sección. La base de datos quedo distribuida de la siguiente manera, en total se tiene 664 tweets de los cuales 201 tienen una orientación guerrerista, 230 tienen una orientación neutra y 233 tienen una orientación pacifista.

## 6.2. CLASIFICACIÓN CON BOLSA DE PALABRAS

A continuación, se muestra el resultado de entrenar los algoritmos con el filtro de bolsa de palabras el cual se implementó en el proyecto porque según lo estudiado en el estado de la cuestión se observó que es una técnica necesaria en el proyecto debido a que los datos son de tipo texto y los algoritmos se deben entrenar con datos numéricos, además al estudiar casos similares se observó que esta técnica es la más utilizada actualmente y brinda buenos resultados.

Ilustración 13. Naive Bayes + Bolsa de palabras

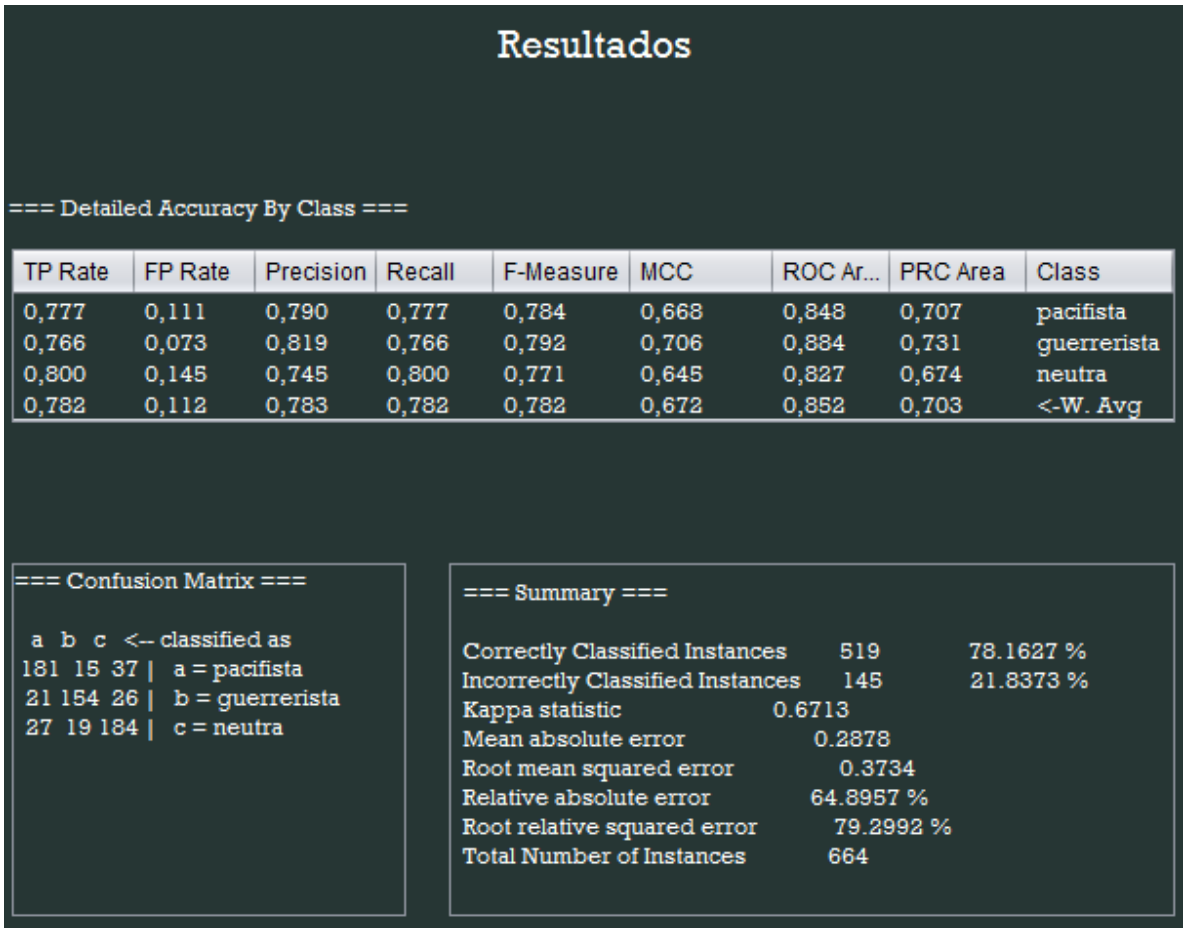



Fuente: Autores.

En la ilustración 13, se observa los resultados obtenidos después del entrenamiento para el algoritmo Naive Bayes en el cual se obtuvo un 65.0602% que corresponde a 432 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.473 que informa que el algoritmo tuvo una buena concordancia diciendo que los datos en su clasificación concuerdan con los

valores predichos, con respecto a la medida ponderada de PRECISION informa que en un 65.4% los valores de dicha clase son realmente de dicha clase, la medida ponderada de RECALL informa que los TP (Verdaderos Positivos) de cada una de las clases se clasificaron en un 65.1% de manera correcta, con la medida ponderada de MCC informa que la calidad de clasificación de las clases fue 0,475 y la medida ponderada de PCR informa que los clasificadores se comportaron en un 73% de manera correcta, se observa la F-Measure (Exactitud) del algoritmo es del 75%.

Ilustración 14. SVM + Bolsa de palabras.

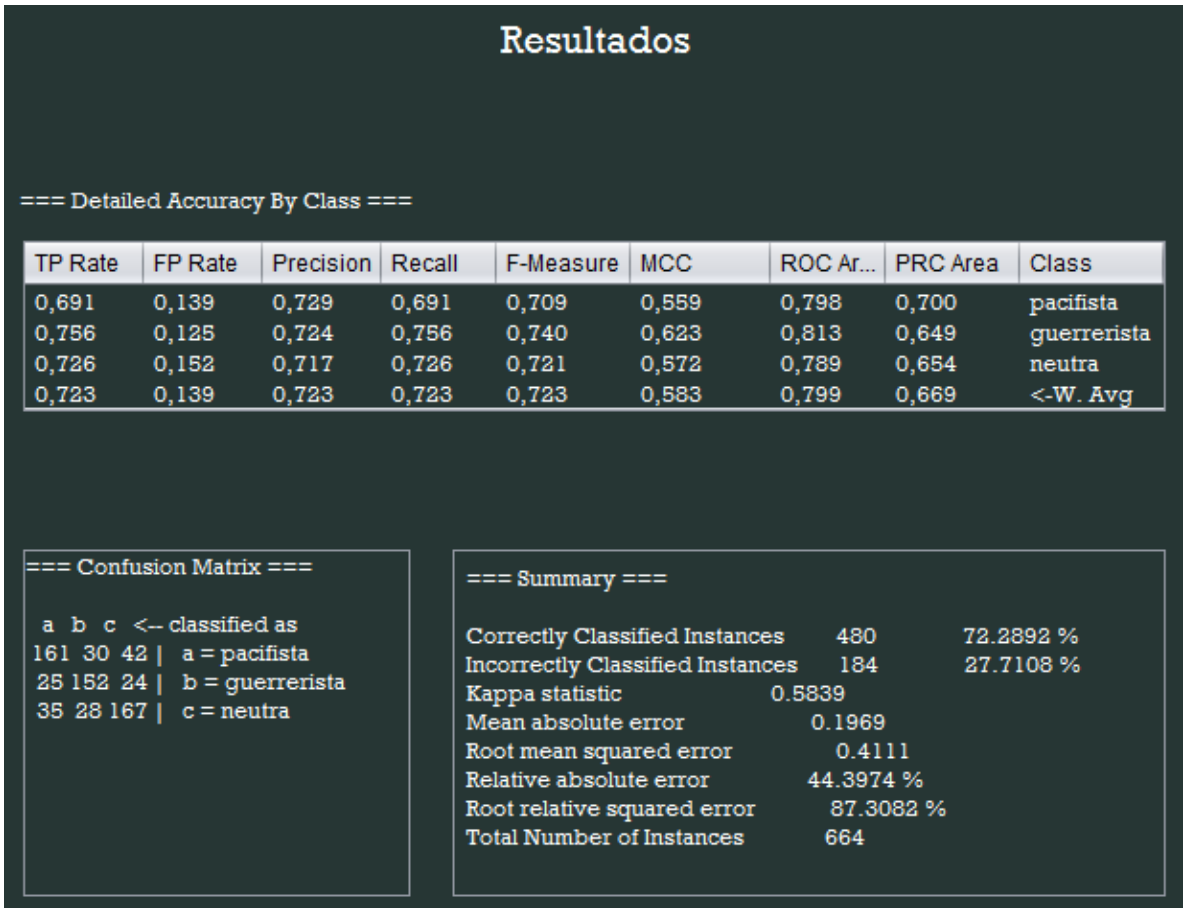



Fuente: Autores.

En la ilustración 14, se observa los resultados obtenidos después del entrenamiento para el algoritmo SVM en el cual se obtuvo un 78.1627% que corresponde a 519 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.6713 que informa que el algoritmo tuvo una buena clasificación de instancias verdaderas y una buena predicción para cada clase, con respecto a la medida ponderada de PRECISION informa que en un 78.3% los valores de cada clase son realmente de dicha clase, la medida ponderada de RECALL

informa que los TP de cada una de las clases se clasificaron en un 78.2% de manera correcta, la exactitud del algoritmo es del 78,2% en sus clasificaciones, con la medida ponderada de MCC informa que la calidad de la clasificación tiende a ser buena y la medida ponderada de PCR informa que los clasificadores se comportaron en un 70.3% de manera correcta.

Ilustración 15. KNN + Bolsa de palabras.

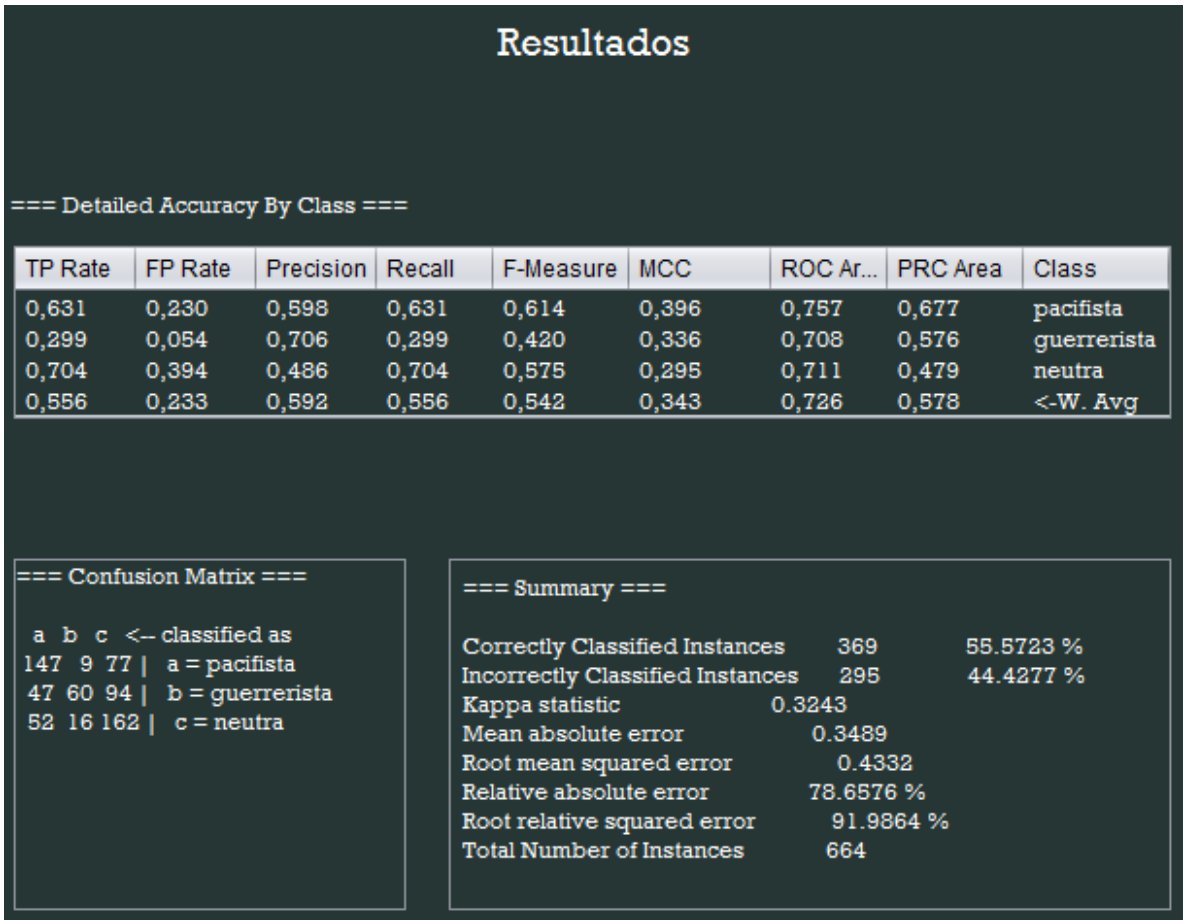



Fuente: Autores.

En la ilustración 15, se observa los resultados obtenidos después del entrenamiento para el algoritmo KNN en el cual se obtuvo un 72.2892% que corresponde a 480 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.5839 que informa que el algoritmo tuvo una buena concordancia entre su clasificación con los valores predichos, con respecto a la medida ponderada de PRECISION informa que en un 72.3% los valores de cada clase son realmente de esa clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 72.3% de manera correcta, la exactitud muestra un balance entre precisión y recall del 72,3% con la medida ponderada de MCC muestra que el 58.3% están relacionados entre sí pero al

observar el MCC dice que la calidad de la clasificación del algoritmo es buena y la medida ponderada de PCR informa que los clasificadores se comportaron en un 66.9% de manera correcta.

Ilustración 16. Red Bayesiana + Bolsa de palabras.

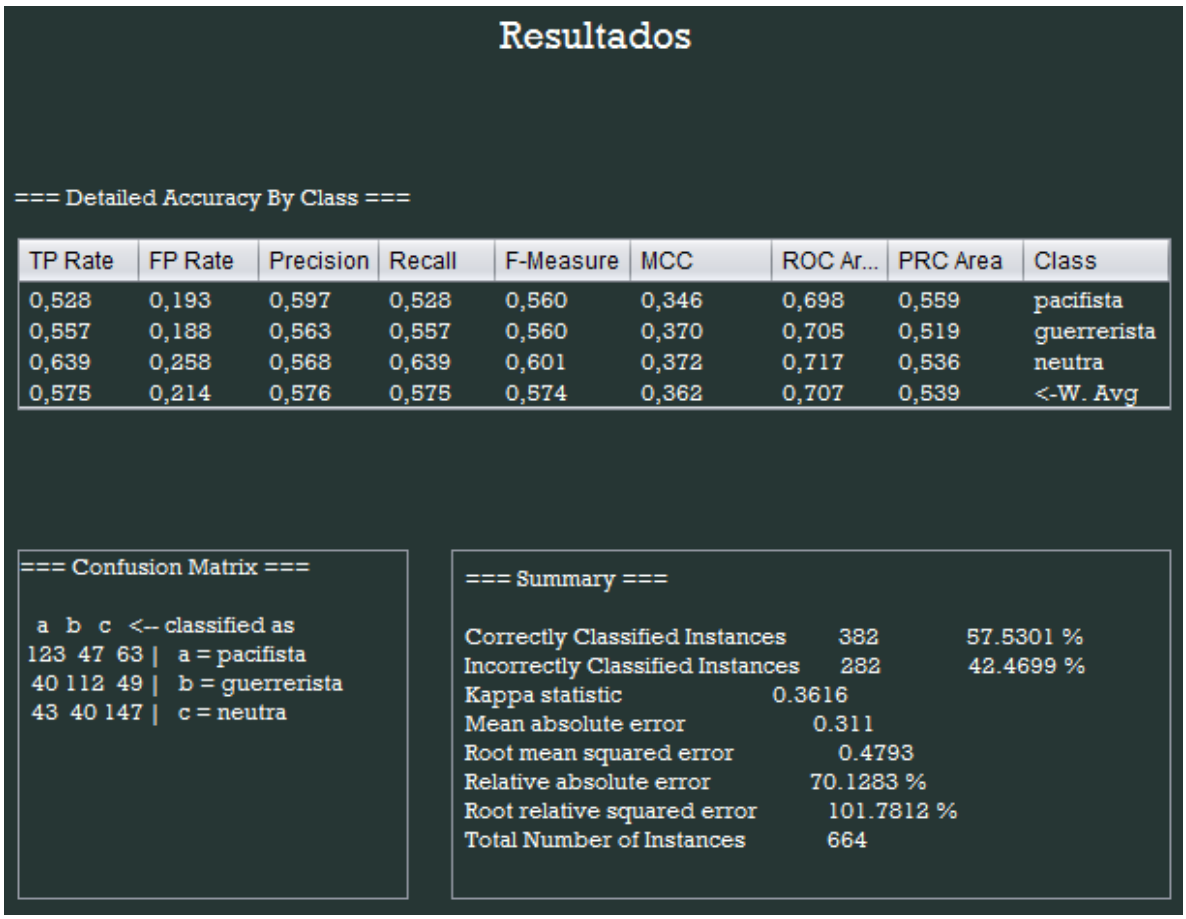



Fuente: Autores.

En la ilustración 16, se observa los resultados obtenidos después del entrenamiento para el algoritmo Red Bayesiana en el cual se obtuvo un 55.5723% que corresponde a 369 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.3243 que informa que el algoritmo tuvo una pobre concordancia entre clasificación de instancias verdaderas y la predicción de las mismas, con respecto a la medida ponderada de PRECISION informa que en un 59.2% los valores de las clases son realmente de esa clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 55.6% de manera correcta, se observa que la exactitud del algoritmo 54,2% con la medida ponderada de MCC informa que la calidad de clasificación es de 34.3% sobre 100% lo que puede ser el causal de la pobre clasificación y la medida

ponderada de PCR informa que los clasificadores se comportaron en un 57.8% de manera correcta.

Ilustración 17. Árbol de decisión + Bolsa de palabras.



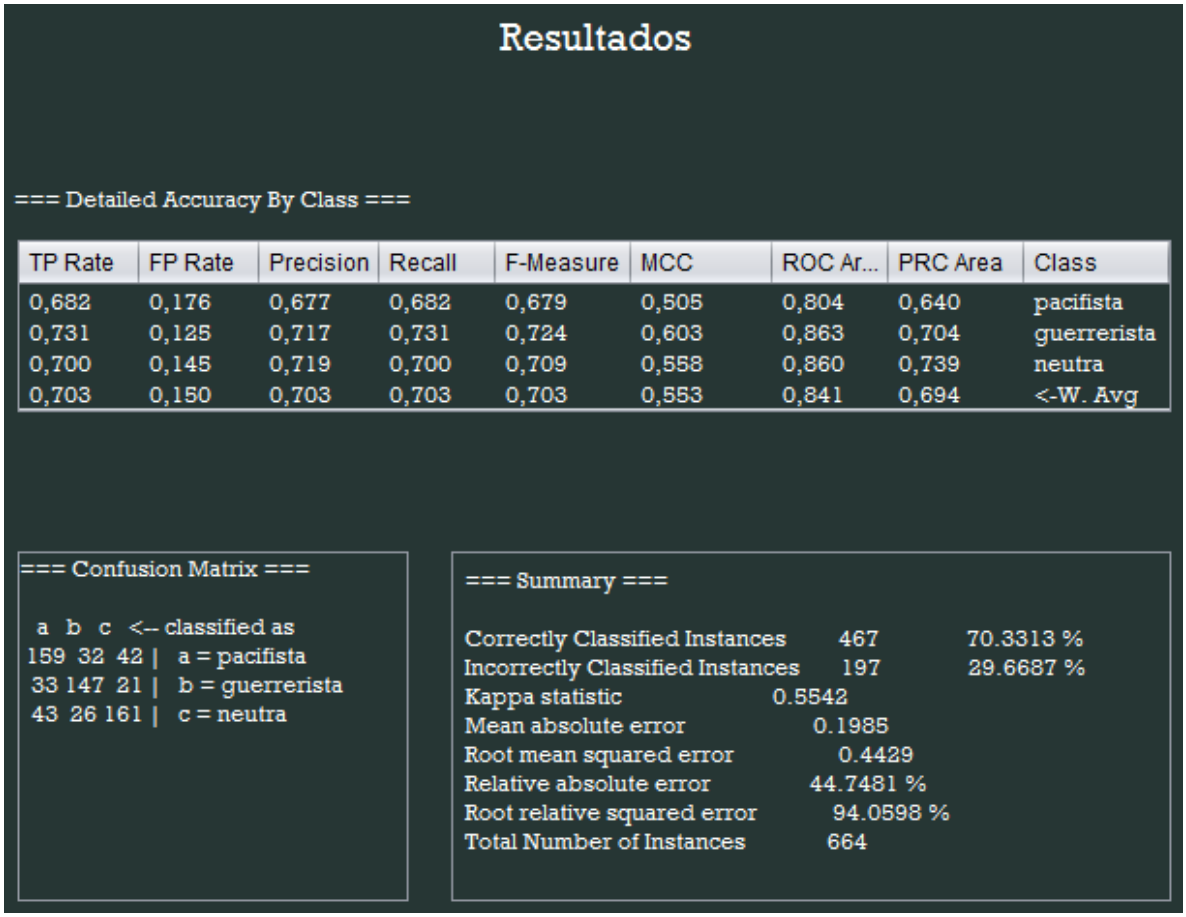

Fuente: Autores.

En la ilustración 17, se observa los resultados obtenidos después del entrenamiento para el algoritmo Árbol de decisión en el cual se obtuvo un 57.5301% que corresponde a 382 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.3616 que informa que el algoritmo tuvo una pobre concordancia entre clasificación de instancias verdaderas y las predichas, con respecto a la medida ponderada de PRECISION informa que en un 57.6% los valores de cada clase son realmente de dicha clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 57.5% de manera correcta, con la medida ponderada de MCC informa que los datos tuvieron una calidad del 36.2% sobre 100%, por eso se tomaron muchas hojas del árbol sin sentido causando que la clasificación y predicción fuese baja y la medida ponderada de PCR informa que los clasificadores se comportaron en un 53.9% de manera correcta.

## 6.3. CLASIFICACIÓN CON BOLSA DE PALABRAS + TF - IDF

A continuación, se muestra el resultado de entrenar los algoritmos con el filtro de bolsa de palabras + TF–IDF el cual se implementó en el proyecto porque, a pesar que la técnica ya fue desarrollada hace algunos años, en la actualidad es la técnica que brinda mejores resultados a nivel de datos en formato de texto, hay otras técnicas que han intentado basarse en el TF-IDF, sin embargo, no se han logrado los resultados que brinda esta técnica, discrimina muy bien las palabras relevantes dentro de un texto lo que hace que le dé una mejor idea al algoritmo de que palabras tomar lo que hace que aumente el porcentaje de clasificación y predicción del mismo.

Ilustración 18. Naive Bayes + Bolsa de palabras + TF-IDF

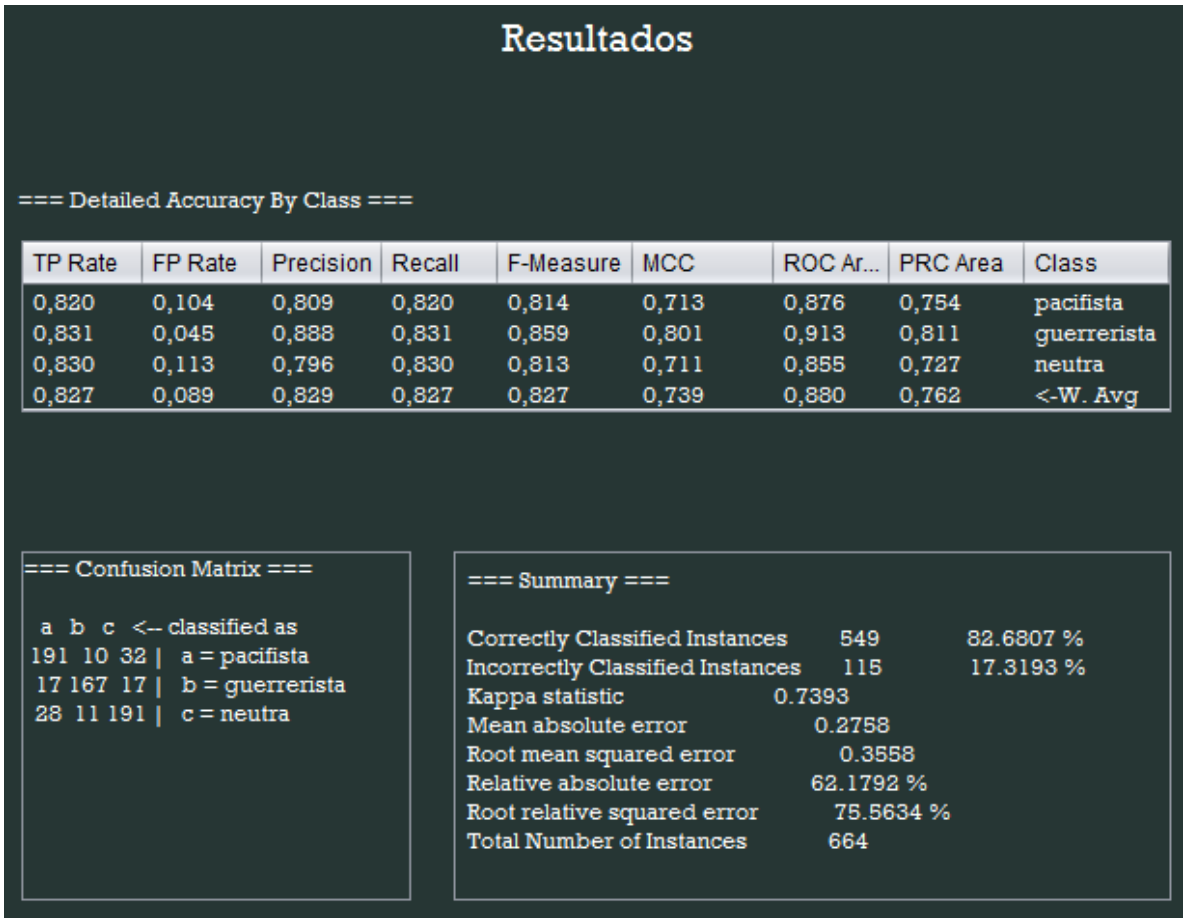



Fuente: Autores.

En la ilustración 18, se observa los resultados obtenidos después del entrenamiento para el algoritmo Naive Bayes en el cual se obtuvo un 70.3313% que corresponde a 467 instancias correctamente clasificadas, con respecto al

valor de Kappa el cual es 0.5542 que informa que el algoritmo tuvo una buena concordancia entre la clasificación de instancias verdaderas con respecto a las predichas, observando la medida ponderada de PRECISION informa que en un 70.3% los valores de dicha clase son realmente pertenecen a esa clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 70.3% de manera correcta, la exactitud del algoritmo es del 70,3%, con la medida ponderada de MCC informa la calidad de clasificación es del 55.3% y la medida ponderada de PCR informa que los clasificadores se comportaron en un 69.4% de manera correcta.

Ilustración 19. SVM + Bolsa de palabras + TF-IDF

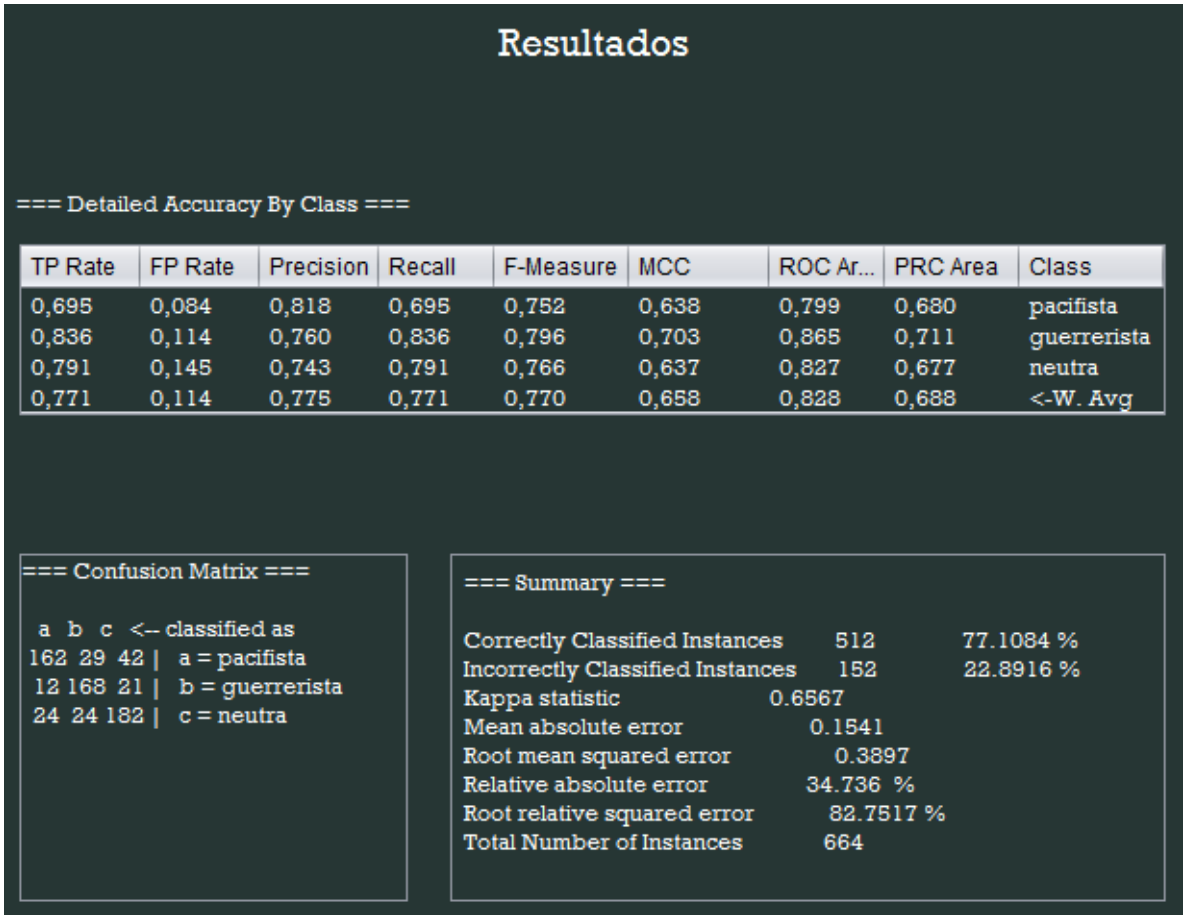



Fuente: Autores.

En la ilustración 19, se observa los resultados obtenidos después del entrenamiento para el algoritmo SVM en el cual se obtuvo un 82.6807% que corresponde a 549 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.7393 que informa que el algoritmo tuvo una buena y casi excelente concordancia entre la clasificación de instancias verdaderas y la

predicción de instancias, con respecto a la medida ponderada de PRECISION informa que en un 82.9% los valores de cada clase pertenecen realmente a dicha clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 82.7% de manera correcta, la exactitud del algoritmo es de 82,7%, con la medida ponderada de MCC se informa que la calidad de clasificación es buena y la medida ponderada de PCR informa que los clasificadores se comportaron en un 76.2% de manera correcta.

Ilustración 20. KNN + Bolsa de palabras + TF-IDF

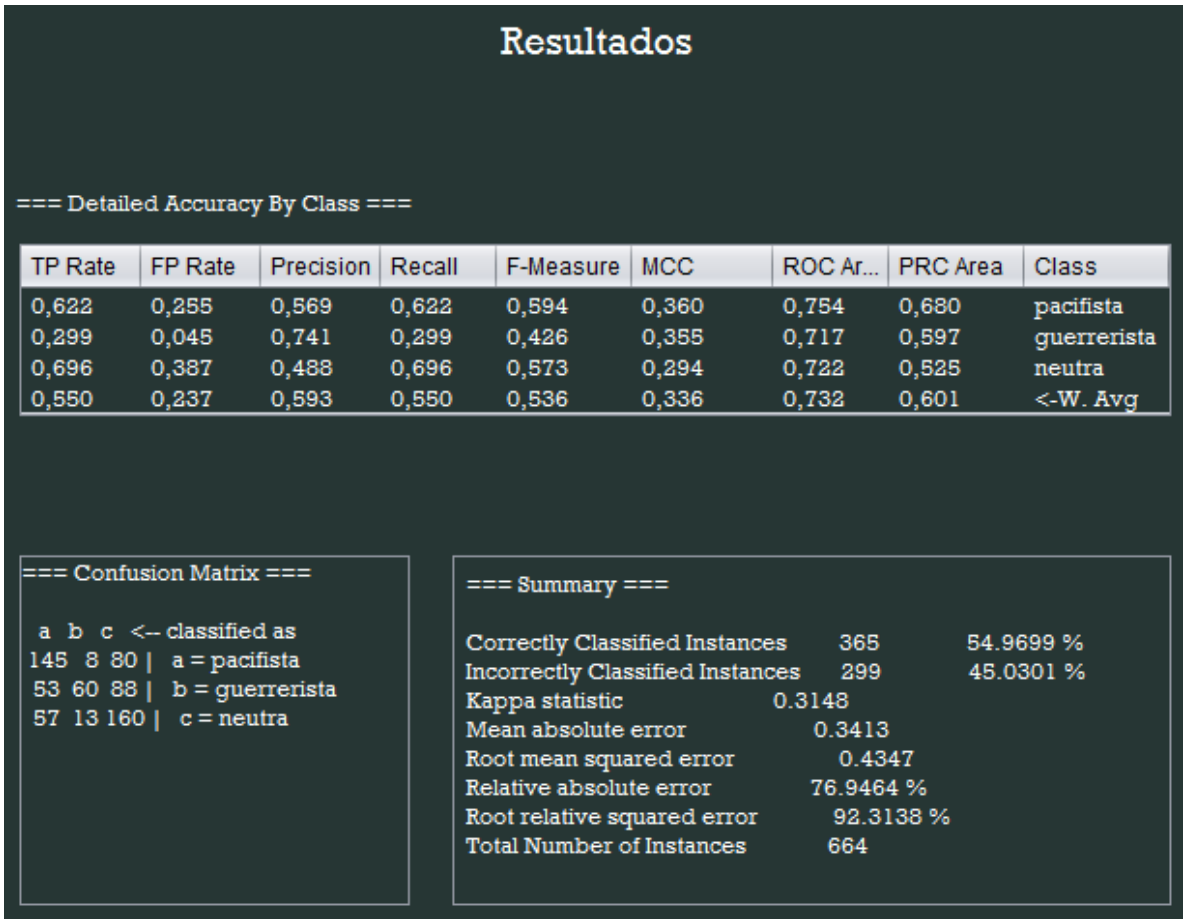



Fuente: Autores.

En la ilustración 20, se observa los resultados obtenidos después del entrenamiento para el algoritmo KNN en el cual se obtuvo un 77.1084% que corresponde a 512 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.6567 que informa que el algoritmo tuvo una buena concordancia entre la clasificación de instancias verdaderas y su predicción, con respecto a la medida ponderada de PRECISION informa que en un 77.5% los valores de cada clase son realmente de dicha clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un

77.1% de manera correcta, la exactitud del algoritmo es del 77% con la medida ponderada de MCC informa que la calidad de la clasificación del algoritmo es buena y la medida ponderada de PCR informa que los clasificadores se comportaron en un 68.8% de manera correcta.

Ilustración 21. Red Bayesiana + Bolsa de palabras + TF-IDF

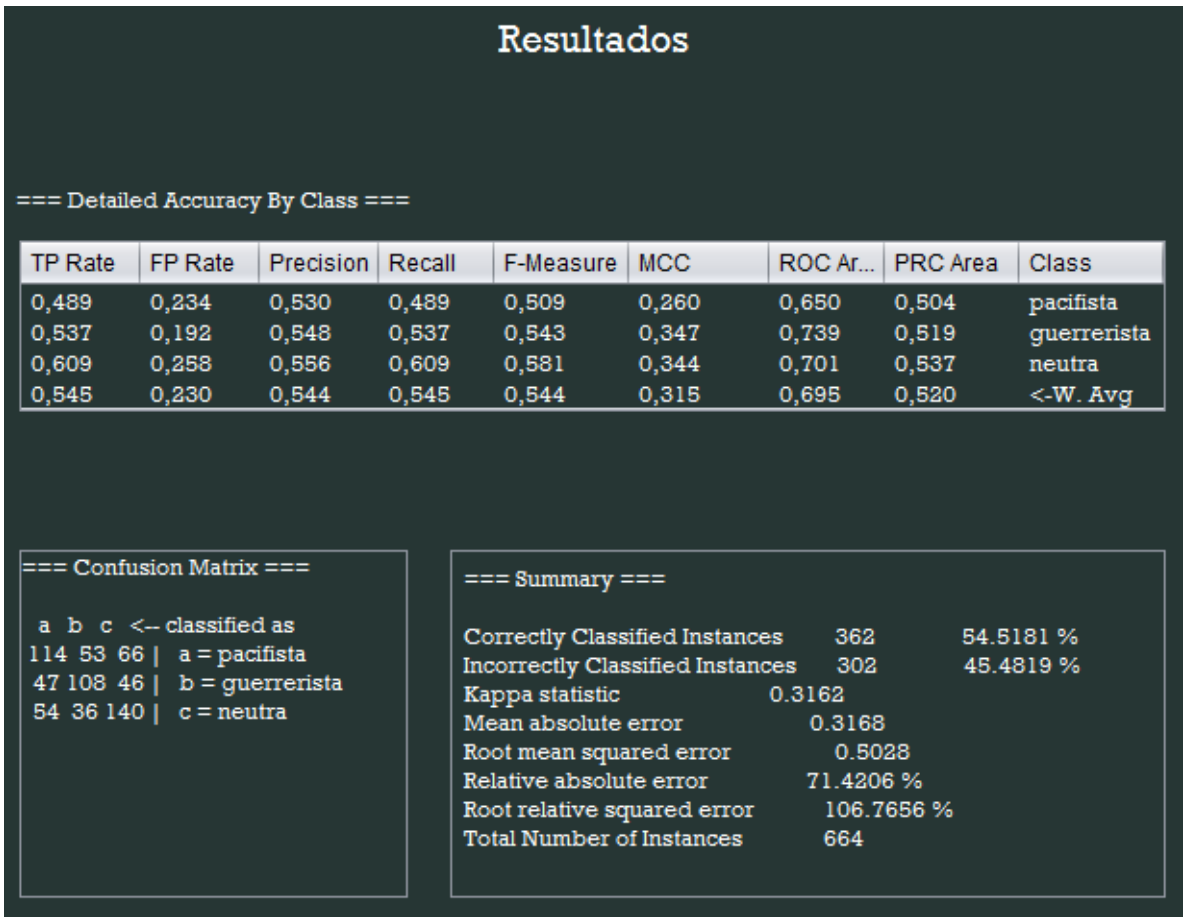



Fuente: Autores.

En la ilustración 21, se observa los resultados obtenidos después del entrenamiento para el algoritmo Red Bayesiana en el cual se obtuvo un 54.9699% que corresponde a 365 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.3148 que informa que el algoritmo no tuvo una buena concordancia entre la clasificación de instancias verdaderas y sus predicciones, con respecto a la medida ponderada de PRECISION informa que en un 59.3% los valores de cada clase son realmente de dicha clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 55% de manera correcta, la exactitud del algoritmo fue de 53,6%, con la medida ponderada de MCC dice que con 33.6% la clasificación del algoritmo es baja, la

medida ponderada de PCR informa que los clasificadores se comportaron en un 60.1% de manera correcta.

Ilustración 22. Árbol de decisión + Bolsa de palabras + TF-IDF




Fuente: Autores.

En la ilustración 22, se observa los resultados obtenidos después del entrenamiento para el algoritmo Árbol de decisión en el cual se obtuvo un 54.5181% que corresponde a 362 instancias correctamente clasificadas, con respecto al valor de Kappa el cual es 0.3162 que informa que el algoritmo tuvo una concordancia pobre entre la clasificación de instancias verdaderas y la predicción de las mismas, con respecto a la medida ponderada de PRECISION informa que en un 54.4% los valores de cada clase son realmente dicha clase, la medida ponderada de RECALL informa que los TP de cada una de las clases se clasificaron en un 54.5% de manera correcta, la exactitud del algoritmo en su clasificación fue de 54,4%, con la medida ponderada de MCC dice que la calidad del algoritmo es baja y la medida ponderada de PCR informa que los clasificadores se comportaron en un 52% de manera correcta en su clasificación.

Se resume los resultados obtenidos después del entrenamiento en la siguiente tabla, se trae a mención los valores más relevantes:

Tabla 8. Resultados del entrenamiento.

| Algoritmo \ | Filtro | Bosa de Palabras | B.P. + TF – IDF |
|---|---|---|---|
| Naive Bayes | Instancias clasificadas correctamente | 65,0602% | 70,3313% |
| | Kappa | 0,473 | 0,5542 |
| | MAE | 0,2435 | 0,1985 |
| | RMSE | 0,4148 | 0,4429 |
| | RRSE | 88,08% | 94,0598% |
| | F - Measure | 0,65 | 0,703 |
| | MCC | 0,475 | 0,553 |
| SVM | Instancias clasificadas correctamente | 78,1627% | 82,6807% |
| | Kappa | 0,6713 | 0,7393 |
| | MAE | 0,2878 | 0,2758 |
| | RMSE | 0,3734 | 0,3558 |
| | RRSE | 79,2992% | 75,5634% |
| | F - Measure | 0,782 | 0,827 |
| | MCC | 0,672 | 0,739 |
| KNN | Instancias clasificadas correctamente | 72,2892% | 77,1084% |
| | Kappa | 0,5839 | 0,6567 |
| | MAE | 0,1962 | 0,1541 |
| | RMSE | 0,4111 | 0,3897 |
| | RRSE | 87,3082% | 82,7517% |
| | F - Measure | 0,723 | 0,77 |
| | MCC | 0,583 | 0,658 |
| Red Bayesiana | Instancias clasificadas correctamente | 55,5723% | 54,9699% |
| | Kappa | 0,3243 | 0,3148 |
| | MAE | 0,3489 | 0,3413 |
| | RMSE | 0,4332 | 0,4347 |
| | RRSE | 91,9864% | 92,3138% |

| | | | |
|---|---|---|---|
| | F - Measure | 0,542 | 0,536 |
| | MCC | 0,343 | 0,336 |
| Árbol de decisión | Instancias clasificadas correctamente | 57,5301% | 54,5148% |
| | Kappa | 0,3616 | 0,3162 |
| | MAE | 0,311 | 0,3168 |
| | RMSE | 0,4793 | 0,5028 |
| | RRSE | 101,7812% | 106,7656% |
| | F - Measure | 0,574 | 0,544 |
| | MCC | 0,362 | 0,315 |

Fuente: Autores.

En la tabla 8, se puede validar qué algoritmo es el mejor para el caso de estudio propuesto y para realizar esta acción, se debe analizar todo el conjunto de datos expuesto, en donde como primera medida se puede observar que el mejor algoritmo es SVM con el filtro de bolsa de palabras + TF – IDF, esto por qué el algoritmo obtuvo el 82.6807% de instancias correctamente clasificadas, con un valor de Kappa 0,7393 el cual informa que el algoritmo presento una excelente clasificación, el valor de MAE y RMSE informan la magnitud promedio de los errores 0,2758 y 0,3558 respectivamente e informa que el algoritmo no presento gran cantidad de errores en la clasificación, se puede ratificar con el valor de RRSE que es de 75,5634%, este valor debe estar lo más cercano a 0 posible y con esto informa el ajuste de los datos, los valores de instancias correctamente clasificadas y el error RRSE están relacionadas directamente, el conjunto de estos valores el algoritmo que mejor se desempeño fue el mencionado SVM seguido de SVM con el filtro de bolsa de palabras.

Para el caso de estudio se puede concluir que el algoritmo que mejor se desempeña es SVM, con la diferencia que con el filtro de bolsa de palabras + TF – IDF es mejor que con solo el filtro de bolsa de palabras, aunque tengan valores muy similares las instancias clasificadas correctamente en conjunto con el valor de Kappa y RRSE es superior SVM con el filtro de bolsa de palabras + TF – IDF, de igual manera el valor de F – Measure informa que el algoritmo seleccionado como el mejor predijo bien y su calidad o MCC también fue alta, lo que ratifica esta decisión.

La conjugación de los resultados para el algoritmo de KNN con el filtro de bolsa de palabras + TF – IDF, no supera al algoritmo de SVM y esto se puede evidenciar porque ningún valor lo supera, excepto el RRSE que es es la raíz cuadrada de la media de errores al cuadrado, en donde este valor debe estar cerca de cero.

Se evidencia que los algoritmos con los peores rendimientos para el presente caso de estudio son la red bayesiana y el árbol de decisión más específicamente al aplicarles el filtro de bolsa de palabras + TF – IDF, se destaca que el error RRSE de árbol de decisión supero el 100%.

# 7. CONCLUSIONES

1. En el presente trabajo de grado se clasificaron textos cortos (tweets), extraídos de la red social twitter, en una orientación guerrerista, pacifista o neutra para un caso de estudio en el conflicto armado colombiano, teniendo en cuenta la aplicación del aprendizaje automático (machine learning).

2. Para la clasificación de textos cortos fue importante redactar un estado de la cuestión mediante revisión sistemática de la literatura en los ámbitos de las técnicas de pre-procesamiento, procesamiento del lenguaje natural y algoritmos de clasificación de textos cortos ya que permitió tener en cuenta casos de estudio similares, también tener el conocimiento de las técnicas que se pueden usar en conjunto y conocer las técnicas y algoritmos más acordes con el caso de estudio del proyecto.

3. Se implementó una interfaz gráfica en el lenguaje de programación Java donde previamente se identificaron distintos entornos o plataformas que permitiesen la clasificación de textos cortos y la integración con el lenguaje de programación, también se identificaron los algoritmos presentes en dichas plataformas y acorde a lo planteado en la revisión sistemática, se comparó cuales algoritmos estaban presentes en las plataformas y en la revisión sistemática, esto permitió elegir los algoritmos a implementar en la interfaz gráfica.

4. La interfaz gráfica mostró resultados de cada uno de los algoritmos implementados, estos resultados permitieron decir que el algoritmo que brinda una mejor respuesta en la clasificación de comunicados de texto corto emitidos por actores del conflicto armado colombiano y extraídos de la red social Twitter, es SVM “Maquinas de soporte vectorial”, debido a que tuvo un resultado de 549 instancias correctamente clasificadas que corresponden al 82.6807% del total de la base de datos.

# 8. TRABAJO FUTURO

- Se puede implementar este proyecto de grado, utilizando los algoritmos y las técnicas de clasificación, para realizar la clasificación de textos cortos en tiempo real y así saber la orientación que tiene un tweet publicado por un actor del conflicto armado colombiano.

- Se puede aplicar el caso de estudio a un escenario que no sea del conflicto armado colombiano, si no que pertenezca a problemáticas sociales o saber la orientación que puede tener un comunicado corto publicado por cualquier persona natural, teniendo en cuenta que se tiene que modificar la base de datos para cumplir el objetivo del proyecto que se centraría en obtener una clasificación correcta de los datos superior al 90% sobre el total de los datos.

- Se puede complementar este proyecto al tratar de proveerle un contexto al algoritmo con la finalidad de que este, al analizar y procesar la base de datos que se le provee este entienda el vocabulario de manera más rápida y eficiente lo que aumentaría considerablemente la eficiencia de los algoritmos en general.

## REFERENCIAS


Adam Marcus y Eugene Wu. (2012). *Text Processing Overview*.

Arcila-Calderón, C., Barbosa-Caro, E., & Cabezuelo-Lorenzo, F. (2016). Técnicas big data: análisis de textos a gran escala para la investigación científica y periodística. *El Profesional de La Información*, *25*(4), 623. https://doi.org/10.3145/epi.2016.jul.12

Bakeman, R., McArthur, D., Quera, V., & Robinson, B. F. (2014). Kappa. *Psychological Methods*, *2*(4), 357–370. https://doi.org/10.1037/1082-989X.2.4.357

Barve, A., Rahate, M., Gaikwad, A., & Patil, P. (2018). Terror Attack Identifier: Classify using KNN, SVM, Random Forest algorithm and alert through messages. *International Research Journal of Engineering and Technology*, 4. Retrieved from www.irjet.net

Bifet, A., & Frank, E. (n.d.-a). *Sentiment Knowledge Discovery in Twitter Streaming Data*. Retrieved from https://www.cs.waikato.ac.nz/ml/publications/2010/Twitter-crc.pdf

Bifet, A., & Frank, E. (n.d.-b). *Sentiment Knowledge Discovery in Twitter Streaming Data*.

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2016). *WEKA Manual for Version 3-8-0*.

Cambronero, C. G., & Moreno, I. G. (2010). *ALGORITMOS DE APRENDIZAJE: KNN & KMEANS*. Madrid. Retrieved from http://blogs.ujaen.es/barranco/wp-content/uploads/2012/02/Algoritmos-de-aprendizaje-knn-y-kmeans.pdf

Centro Nacional de Información de Ciencias Médicas., E., & Cabrera-Gato, J. E. (2007). *Minería de textos: Una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital. ACIMED* (Vol. 16). 2000, Editorial Ciencias Médicas. Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352007001000005

Corso, I., & Lorena, C. (2010). *Aplicación de algoritmos de clasificación supervisada usando Weka*.

De Mauro, A., Greco, M., & Grimaldi, M. (2015). What is big data ? A consensual definition and a review of key research topics. *Big Data Comput. Sci. Eng*, *1644*, 9. https://doi.org/10.1063/1.4907823

Del Pilar, A., & Robles, A. (2017). *ANÁLISIS DE EFECTIVIDAD AL IMPLEMENTAR LA TÉCNICA DE ÁRBOLES DE DECISIÓN DEL ENFOQUE*

*DE APRENDIZAJE DE MÁQUINA PARA LA DETERMINACIÓN DE AVALÚOS MASIVOS PARA LAS UPZ 79 CALANDAIMA, 65 ARBORIZADORA Y 73 GARCÉS NAVAS*. Retrieved from http://repository.udistrital.edu.co/bitstream/11349/5779/1/AlbancandoRoblesAdrianaDelPilar2017.pdf

Diccionario Cambridge. (2018). Tweet.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K. (n.d.). *Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification*. Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P14-2009

Escortell Pérez, A., Giménez Fayos, M., & Rosso, P. (2017). The Impact of Emotions on Polarity Analysis using Figurative Language in Twitter, 8. Retrieved from http://alt.qcri.org/semeval2015/task11/

Figuerola, C. G., Alonso Berrocal, J. L., Zazo Rodríguez, Á. F., & Rodríguez, E. (2004). Algunas Técnicas de Clasificación Automática de Documentos. *Página*, *15*, 10. Retrieved from https://core.ac.uk/download/pdf/153334293.pdf

Frank, E., Hall, M. A., Witten, I. H., & Kaufmann, M. (2016a). *WEKA Workbench Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"* Retrieved from https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

Frank, E., Hall, M. A., Witten, I. H., & Kaufmann, M. (2016b). *WEKA Workbench Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*

Gálvez, C. (2008). MINERÍA DE TEXTOS: LA NUEVA GENERACIÓN DE ANÁLISIS DE LITERATURA CIENTÍFICA EN BIOLOGÍA MOLECULAR Y GENÓMICA, 14. Retrieved from https://doi.org/10.5007/1518-2924.2008v13n25p1

Hosmer, D. W., & Lemeshow Stanley. (1989). *Applied logistic regression*. Retrieved from http://resource.heartonline.cn/20150528/1_3kOQSTg.pdf

KDnuggets. (2017). A General Approach to Preprocessing Text Data. Retrieved September 19, 2018, from https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html

Kuhn, M. (2018). The caret Package. Retrieved September 3, 2018, from http://topepo.github.io/caret/index.html

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). *From Word Embeddings To Document Distances*. Retrieved from http://proceedings.mlr.press/v37/kusnerb15.pdf

Le, Q., & Mikolov, T. (2014). *Distributed Representations of Sentences and*

*Documents*. Retrieved from https://cs.stanford.edu/~quocle/paragraph_vector.pdf

Letelier, P., & Penadés, M. C. (2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Retrieved from www.agileuniverse.com.

Levy, O., Goldberg, Y., & Dagan, I. (2015). *Improving Distributional Similarity with Lessons Learned from Word Embeddings*. Ramat Gan, Israel. Retrieved from https://www.transacl.org/ojs/index.php/tacl/article/view/570

Los Modelos Geométricos (Modelos parte I). (n.d.). Retrieved September 30, 2018, from https://mlmexicanguy.wordpress.com/2017/02/09/los-modelos-geometricos-modelos-parte-i/

Miguel Ángel Vallejo Pareja. (2006). *MINDFULNESS*. Retrieved from http://www.redalyc.org/html/778/77827204/

Modelos Logicos (Modelos parte 3). (n.d.). Retrieved September 30, 2018, from https://mlmexicanguy.wordpress.com/2017/02/23/modelos-logicos-modelos-parte-3/

Modelos Probabilísticos (Modelos parte 2). (n.d.). Retrieved September 30, 2018, from https://mlmexicanguy.wordpress.com/2017/02/12/modelos-probabilisticos-modelos-parte-2/

Pascual, D., Pla, F., & Sánchez, S. (n.d.). *Algoritmos de agrupamiento*. Retrieved from http://marmota.dlsi.uji.es/WebBIB/papers/2007/1_Pascual-MIA-2007.pdf

Patil, T. R., & Sherekar, M. S. S. (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications*, *6*(2), 256–261. Retrieved from http://www.cs.bme.hu/~kiskat/adatb/bank-data-

Paute, D. J., Soroa, A., & López, O. (2016). *ANÁLISIS Y CLASIFICACIÓN DE INFORMACIÓN MEDIÁTICA ELECTORAL UTILIZANDO MINERÍA DE TEXTO*. Retrieved from https://addi.ehu.es/bitstream/handle/10810/19300/TesisFinal_26092016.pdf?sequence=1&isAllowed=y

Pérez Abelleira, A., & Cardoso, C. A. (2010). *Minería de texto para la categorización automática de documentos* (Vol. 5). Salta, Argentina. Retrieved from http://www.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p11-alicia-articulo-cuadernos-formateado.pdf

Radio, C. (2017). De las balas a los votos. Retrieved from http://caracol.com.co/radio/2017/11/24/nacional/1511492119_078663.html

Rangra, K., & Bansal Research Scholar Professor, K. L. (2014). *Comparative Study of Data Mining Tools*. *International Journal of Advanced Research in*

*Computer Science and Software Engineering* (Vol. 4). Retrieved from www.r-project.org

RapidMiner. (2014). *RapidMiner Studio Manual*. Retrieved from https://docs.rapidminer.com/downloads/RapidMiner-v6-user-manual.pdf

Ratinov, L., & Roth, D. (2009). *Design Challenges and Misconceptions in Named Entity Recognition*. Urbana, USA. Retrieved from http://l2r.cs.uiuc.edu/

Repaso didáctico sobre machine learning. (n.d.). Retrieved September 30, 2018, from https://lapastillaroja.net/2015/02/ml-algols/

Reyes-Ortiz, J. A., Paniagua-Reyes, F., & Sánchez, L. (2017). *Mining of Opinions Centered on Topics Using Short Texts in Spanish*. *Research in Computing Science* (Vol. 134). Ciudad de México, México. Retrieved from http://www.rcs.cic.ipn.mx/rcs/2017_134/Mineria de opiniones centrada en topicos usando textos cortos en espanol.pdf

Ritter, A., Clark, S., & Etzioni, O. (2011). *Named Entity Recognition in Tweets: An Experimental Study*. Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/D11-1141

Rosenthal, S., Farra, N., & Nakov, P. (n.d.). *SemEval-2017 Task 4: Sentiment Analysis in Twitter*. Retrieved from https://trends24.in/

Santana Mansilla, P., Costaguta, R., & Missio, D. (2014). Aplicación de Algoritmos de Clasificación de Minería de Textos para el Reconocimiento de Habilidades de E-tutores Colaborativos. *Inteligencia Artificial*, *17*(53), 57–67. Retrieved from http://journal.iberamia.org/

Sasaki, Y., & Fellow, R. (2007). *The truth of the F-measure*.

*scikit-learn user guide*. (2018). Retrieved from http://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf

scikit learn. (2017). Documentation scikit-learn: machine learning in Python — scikit-learn 0.19.2 documentation. Retrieved September 3, 2018, from http://scikit-learn.org/stable/documentation.html

Shashanka, M. (2011). *A FAST ALGORITHM FOR DISCRETE HMM TRAINING USING OBSERVED TRANSITIONS*. East Hartford. Retrieved from http://cns.bu.edu/~mvss/stuff/ShashankaICASSP2011.pdf

Srividhya, V., & Anitha, R. (2010). Evaluating Preprocessing Techniques in Text Categorization. *International Journal of Computer Science and Application Issue*. Retrieved from http://sinhgad.edu/ijcsa-2012/pdfpapers/1_11.pdf

Sucar, L. E. (2011). *Redes Bayesianas*. Retrieved from https://ccc.inaoep.mx/~esucar/Clases-mgp/caprb.pdf

The MathWorks Inc. (2018). MATLAB - El lenguaje del cálculo técnico - MATLAB

& Simulink. Retrieved September 3, 2018, from https://la.mathworks.com/products/matlab.html

Tornero Lucas, J. (2017). *Machine Learning: Modelos Ocultos de Markov (HMM) y Redes Neuronales Artificiales (ANN)*. Barcelona, España. Retrieved from http://diposit.ub.edu/dspace/bitstream/2445/122446/2/memoria.pdf

Universidad Nacional Mayor de San Marcos. Facultad de Ingeniería Industrial. Instituto de Investigación, Oscar; Rosales López, Pedro Pablo; Salas Bacalla, J. (2010). *Criterios de selección de metodologías de desarrollo de software*. *Industrial Data* (Vol. 13). Universidad Nacional Mayor de San Marcos. Retrieved from http://www.redalyc.org/html/816/81619984009/

Valdivia, Eugenio Martínez Cámara, M. Teresa Martín Valdivia, José M. Perea Ortega L, A. U. L. (2011). *Técnicas de clasificación de opiniones aplicadas a un corpus en español*. Retrieved from http://www.booking.com

Venables, W. N., & Smith, D. M. (1997). *An Introduction to R Notes on R: A Programming Environment for Data Analysis and Graphics Version 3.5.1 (2018-07-02)*. *R. Gentleman & R. Ihaka Copyright c*. Retrieved from https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf

Venegas, R. (2007). Clasificación de textos académicos en función de su contenido léxico-semántico. *Revista Signos*, *40*(63), 239–271. https://doi.org/10.4067/S0718-09342007000100012

Xiang, G., Fan, B., Wang, L., Hong, J. I., & Rose, C. P. (2012). *Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus*. Retrieved from http://www.cs.cmu.edu/~binfan/papers/cikm12_twitter.pdf

Zontisa. (2018). El etiquetado gramatical o POS tagging - Zontisa Smart Technology. Retrieved September 2, 2018, from https://www.zontisa.com/el-etiquetado-gramatical-o-pos-tagging/

## ANEXO A: MANUAL DE USUARIO

En el presente anexo se presenta el manual de usuario del programa que muestra la solución realizada para la presente tesis.

UNIVERSIDAD CATÓLICA
de Colombia

# MANUAL DE USUARIO.


Presentado por:
**Juan Pablo Páramo Lozada. Código: 625381**
**Cesar Augusto Espitia Betancourt. Código: 625373**

Asesores:
Raúl Ernesto Menéndez Mora, PhD
Erika Paola Holguín Ontiveros.

Universidad Católica de Colombia
Facultad de ingeniera
Departamento de ingeniería de sistemas
Bogotá Colombia

2018

## TABLA DE CONTENIDO

# 1. INTRODUCCIÓN

La solución desarrollada en el lenguaje de programación java, nos permite resolver la problemática planteada en el documento expuesto para la tesis titulada “APLICACIÓN DEL APRENDIZAJE AUTOMÁTICO EN LA CLASIFICACIÓN DE TEXTOS CORTOS UN CASO DE ESTUDIO EN EL CONFLICTO ARMADO COLOMBIANO”.

A continuación, se muestra cómo utilizar el programa para validar que algoritmo utilizando Machine Learning con aprendizaje supervisado se desempeña mejor en el caso de estudio.

## 1.1. INICIAR EL PROGRAMA

Para iniciar el programa como primer paso se debe abrir la carpeta en la cual se encuentra contenida el programa “tesis” con extensión .jar, seguidamente se debe dar doble click sobre el icono o dar click derecho sobre el icono y seleccionar la opción **abrir,** el icono se puede ver en la siguiente imagen**.**


## 1.2. CERRAR EL PROGRAMA

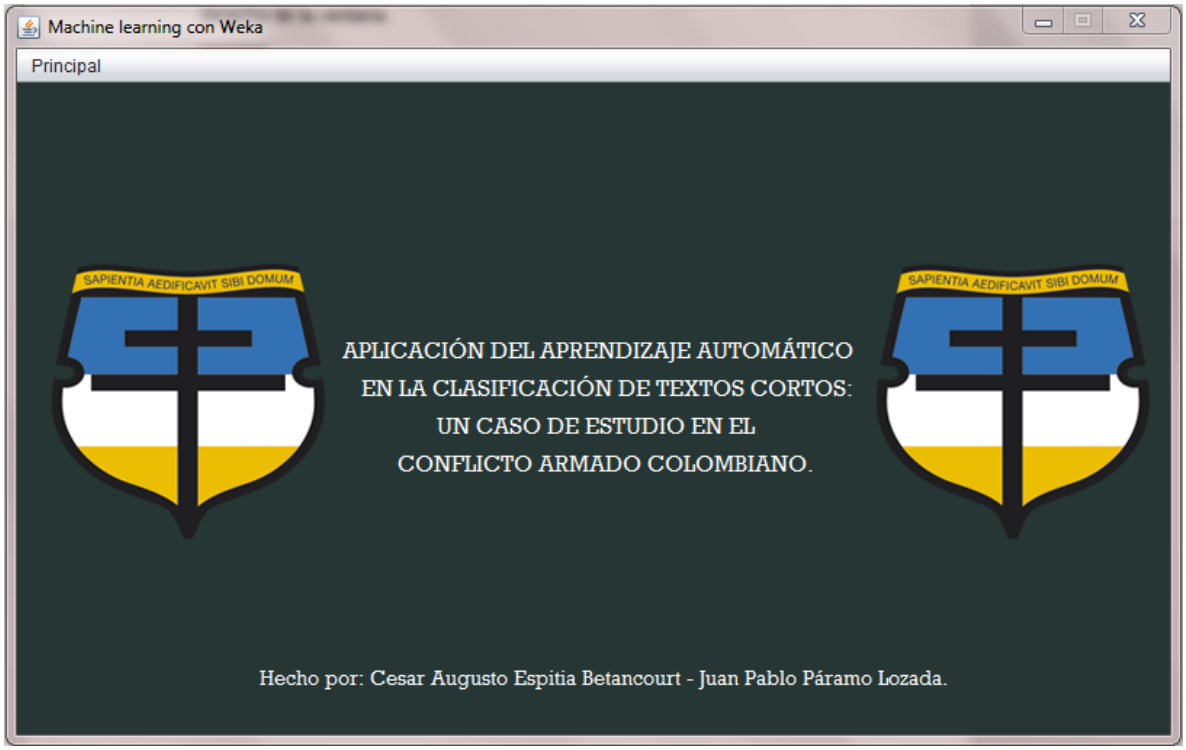
Parra **cerrar** el programa, se pueden utilizar cualquiera de las siguientes opciones:

- Hacer clic en el botón cerrar, este botón se encuentra situado en la parte superior derecha de la ventana.

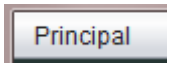
- También se puede pulsar la combinación de teclas **ALT+F4**, con esta combinación de teclas se cerrará la ventana que se tenga activa en ese momento.

## 1.3. PANTALLA INICIAL

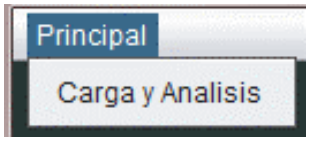
Al iniciar el programa aparecerá una pantalla inicial como la siguiente.




Ampliando el menú de interacción.



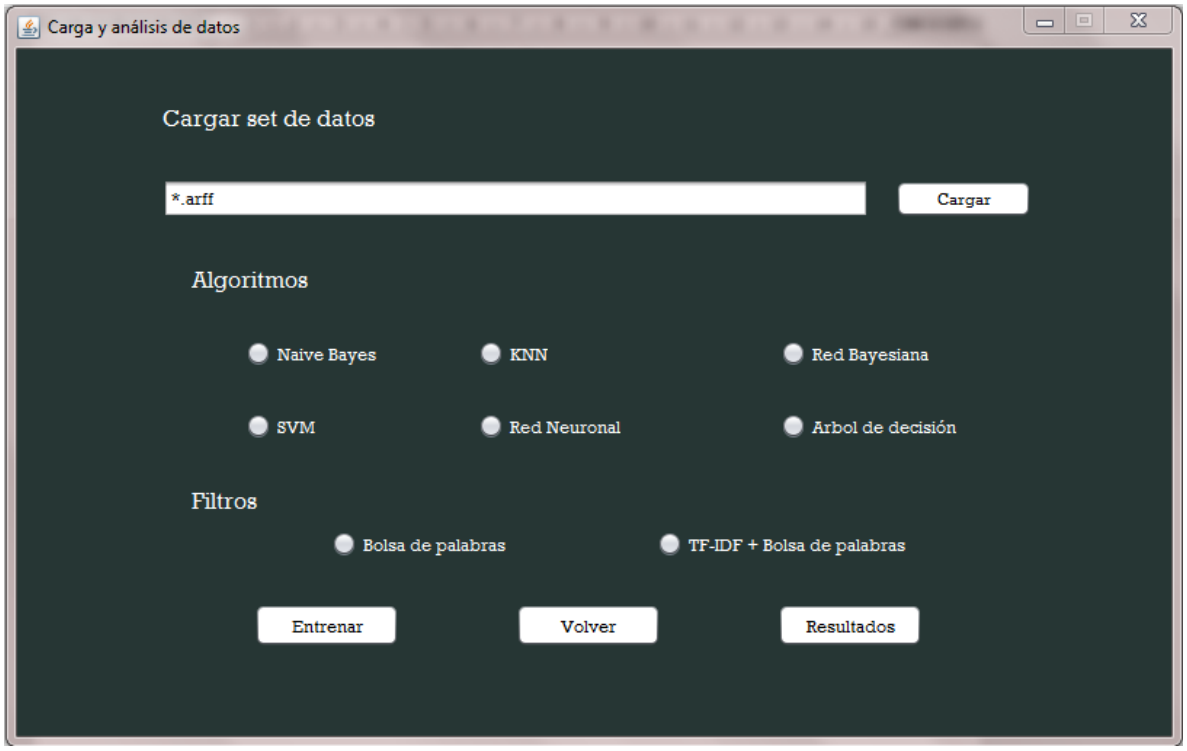

Se puede observar un submenú "**Principal**" el cual al seleccionarlo muestra la siguiente opción.




Para continuar utilizando el programa y validar la solución propuesta se debe seleccionar la opción **Carga y Análisis.**

# 2. UTILIZACIÓN DEL SISTEMA

Una vez seleccionada la opción **Carga y Análisis d**el submenú **Principal,** se muestra la siguiente pantalla:

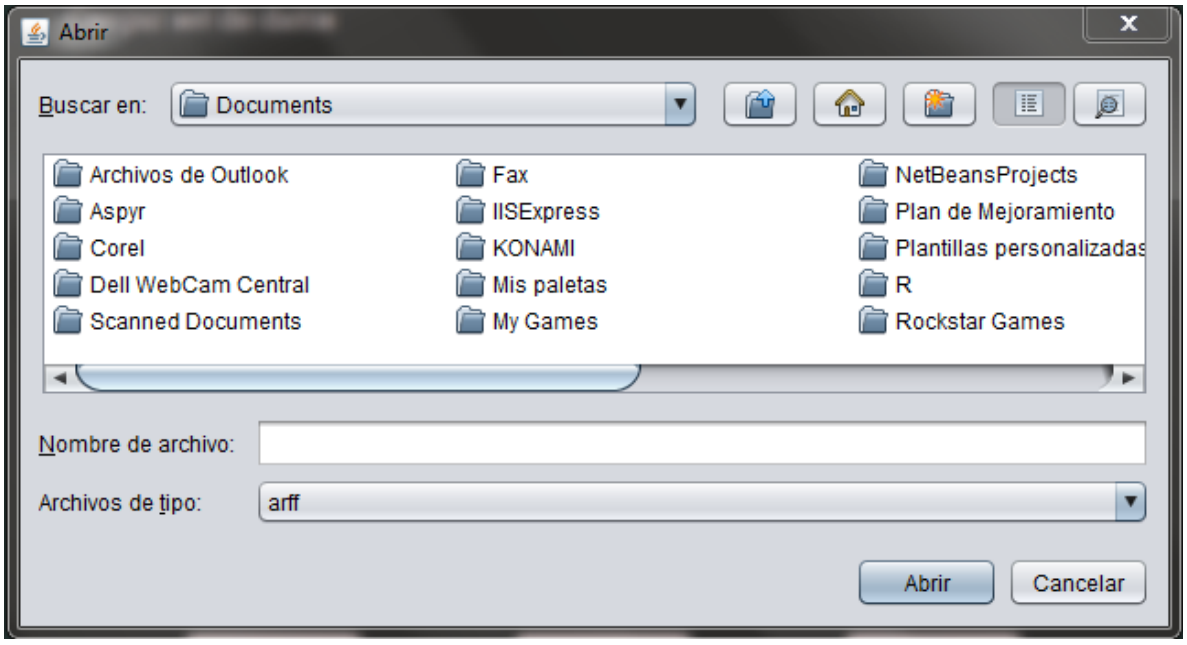



Esta pantalla está dividida en cuatro secciones, las cuales se detallan a continuación.

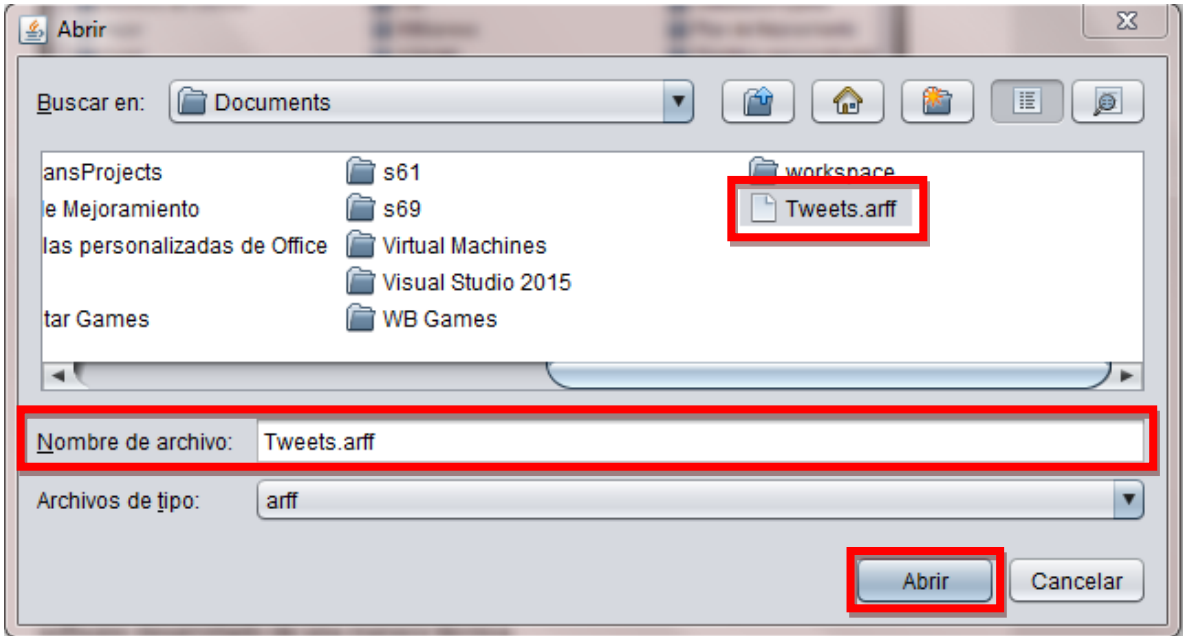
## 2.1. CARGAR

Esta sección tiene como objetivo cargar el conjunto de datos a analizar y se realiza seleccionando el botón de **cargar**, el cual está encerrado en un cuadro en color rojo como se puede observar en la siguiente imagen.


Una vez seleccionado este botón, el programa muestra la siguiente pantalla.



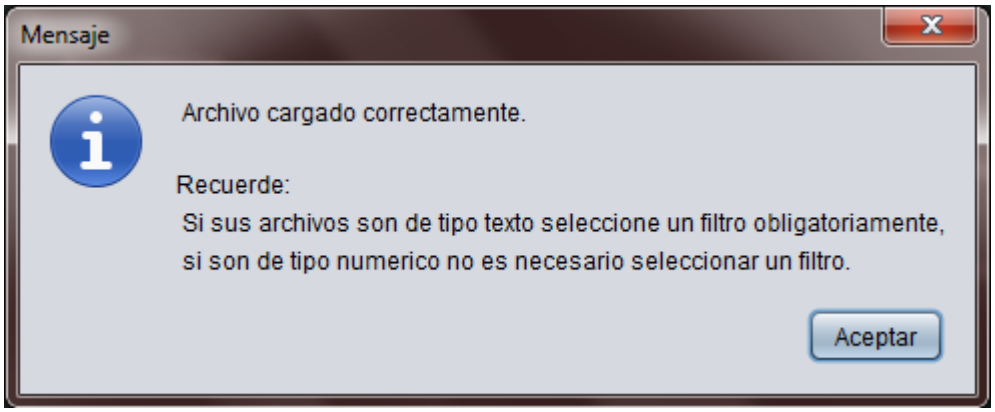

Esta pantalla permite al usuario seleccionar un conjunto de datos para su posterior análisis y como se recomendó en el documento, se debe seleccionar un conjunto de datos con extensión .arff como se puede ver a continuación.



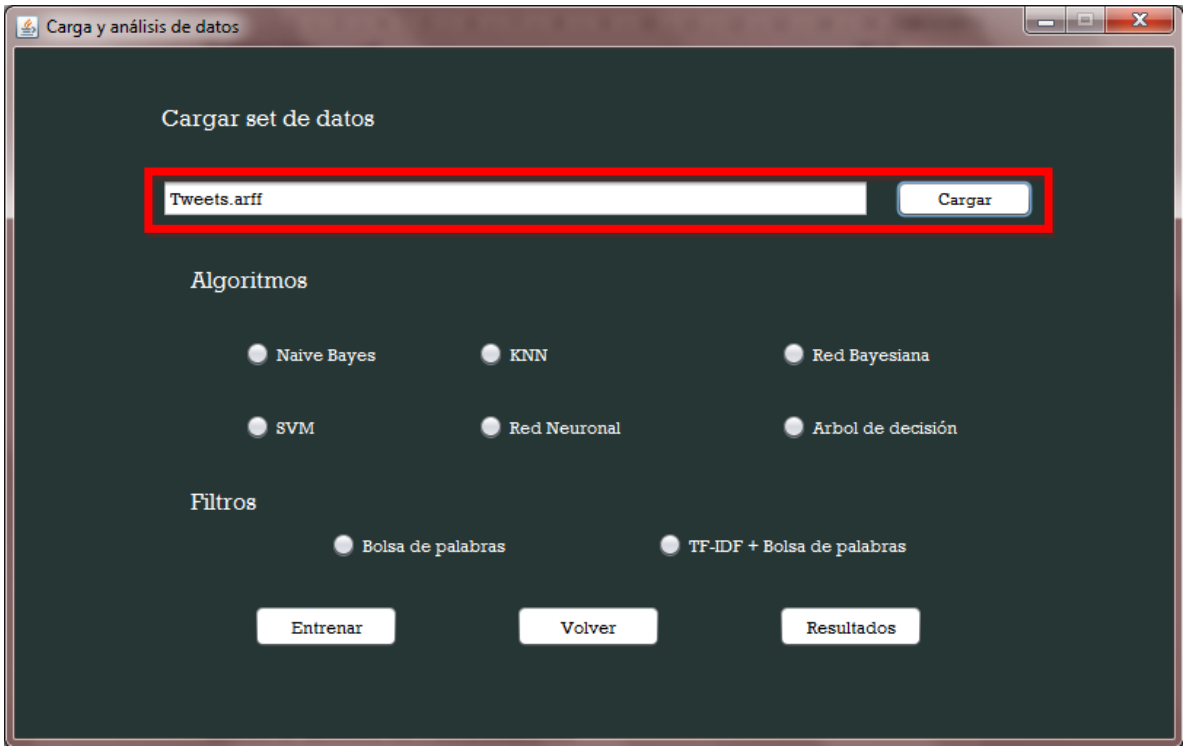

Se puede detallar que en Nombre de Archivo paso de estar en blanco a contener el nombre del conjunto de datos, como se puede observar en la anterior imagen.

Seguidamente de seleccionar el conjunto de datos, se debe seleccionar el botón de **Abrir** para que el programa lo cargue y permita su análisis.

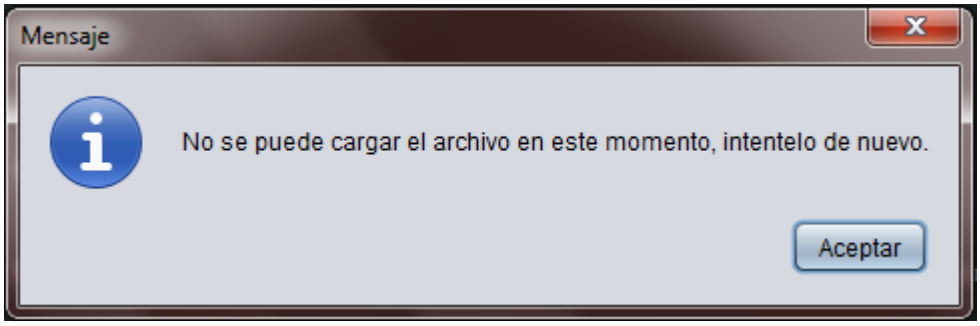
Una vez seleccionado el botón de **Abrir** mostrara la siguiente pantalla.




Cuando se seleccione el botón **Aceptar**, se muestra la pantalla inicial de la segunda sección, pero esta vez con el conjunto de datos cargado en el programa, como se puede ver en la siguiente imagen y en el recuadro encerrado en color rojo.



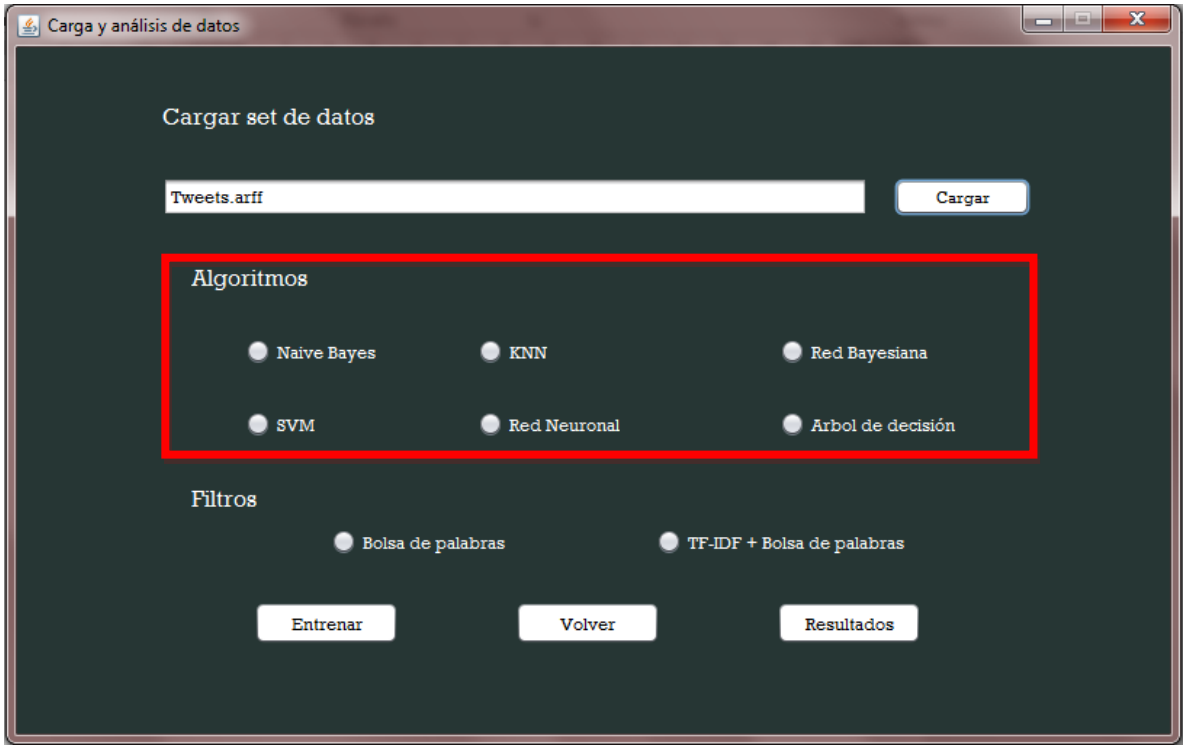
En caso tal de que el programa no pueda cargar el conjunto de datos, muestra la siguiente pantalla.




## 2.2. ALGORITMOS

Esta sección tiene como objetivo seleccionar el algoritmo el cual se aplicará al conjunto de datos cargado.

La sección está encerrada en un cuadro de color rojo como se puede observar en la siguiente imagen.




En esta sección se puede detallar que se encuentran los siguientes algoritmos:

- Naive Bayes
- KNN
- Red Bayesiana
- SVM
- Red Neuronal
- Árbol de decisión

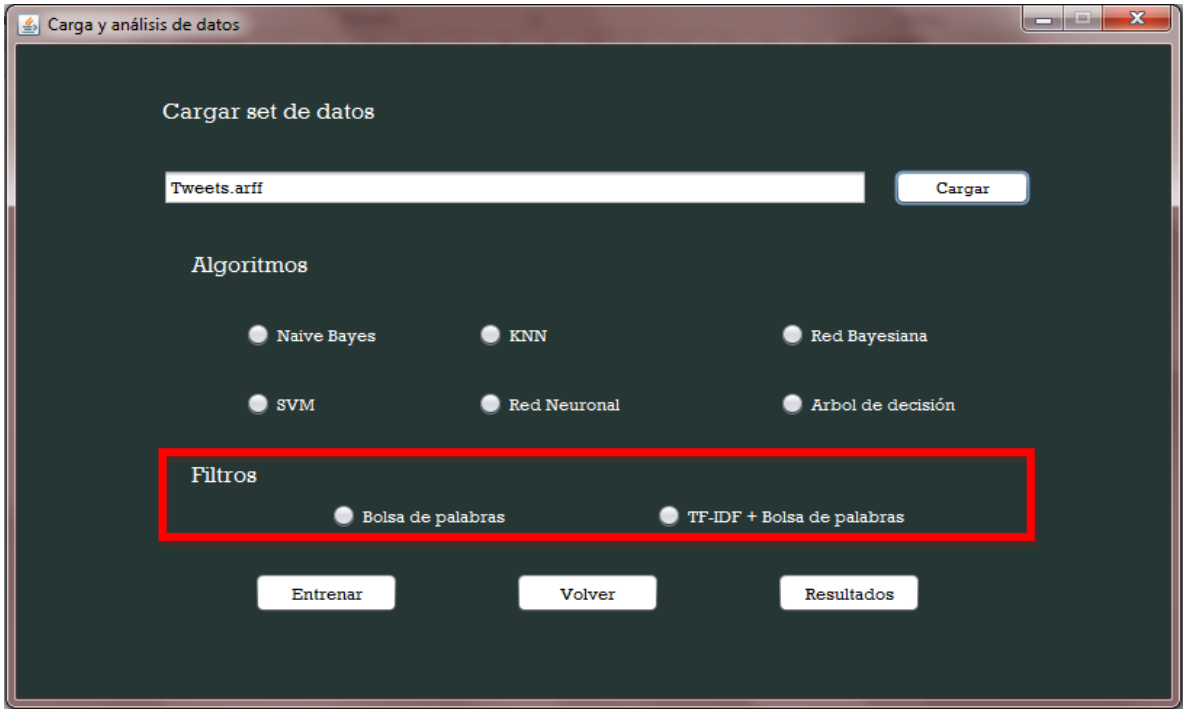
Moviendo el cursor hasta el botón circular que se encuentra al lado del algoritmo, el usuario podrá seleccionarlo, un ejemplo se puede observar en la siguiente imagen.


Si el usuario desea cambiar el algoritmo tiene que seleccionar otro de la misma manera que selecciono el anterior.

## 2.3. FILTROS

Esta sección tiene como objetivo seleccionar un filtro para aplicarle al conjunto de datos cargado.

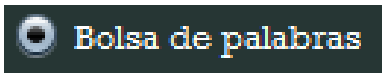
La sección está encerrada en un cuadro de color rojo como se puede observar en la siguiente imagen.



En esta sección se puede detallar que se encuentran los siguientes filtros:

- Bolsa de palabras
- TF – IDF + Bolsa de palabras

Moviendo el cursor hasta el botón circular que se encuentra al lado del filtro, el usuario podrá seleccionarlo, un ejemplo se puede observar en la siguiente imagen.

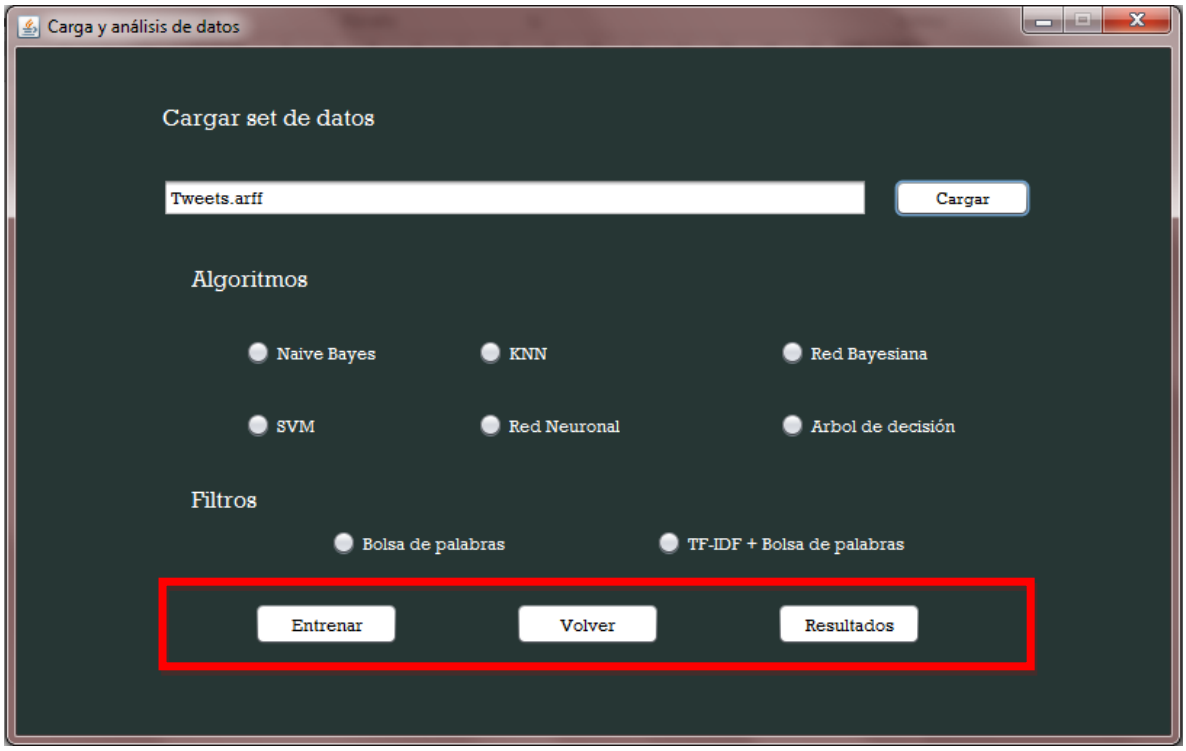



Si el usuario desea cambiar el filtro tiene que seleccionar otro de la misma manera que selecciono el anterior.

## 2.4. BOTONES DE ACCION

Esta sección tiene como objetivo permitirle al programa interactuar según las preferencias del usuario final.

La sección está encerrada en un cuadro de color rojo como se puede observar en la siguiente imagen.



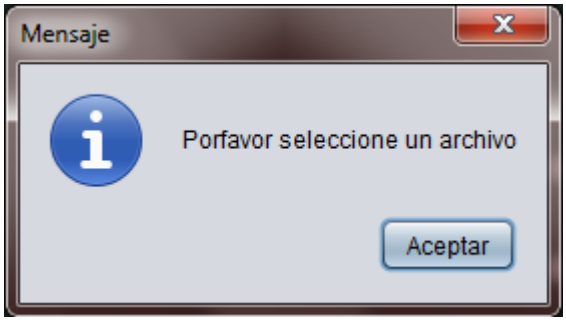

Los botones de acción se detallan a continuación.

### 2.4.1. Entrenar.

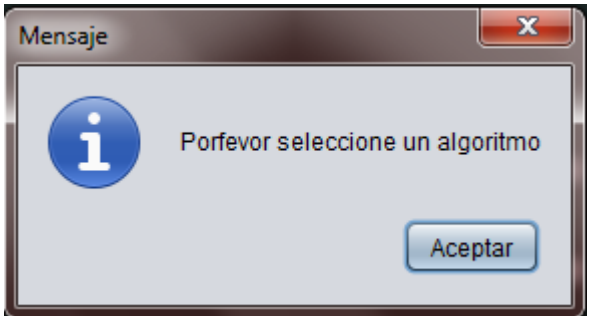
El botón entrenar permite al usuario, como su nombre lo indica, entrenar el algoritmo con el conjunto de datos seleccionado y el filtro seleccionado.

Para que funcione correctamente tiene que estar cargado un conjunto de datos y se debió seleccionar un algoritmo, en cuanto al filtro, si los datos son de tipo numérico no necesariamente se debe seleccionar un filtro, de lo contrario es obligatorio seleccionarlo.

En caso tal de que no se haya cargado un conjunto de datos por parte del usuario, se muestra una pantalla como la siguiente.

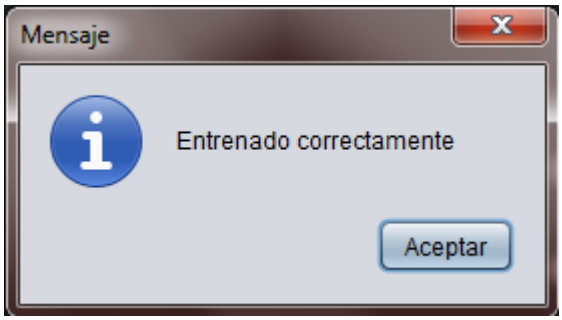



En caso tal de que no se haya seleccionado un algoritmo por parte del usuario, se muestra una pantalla como la siguiente.




Si el usuario cargo el conjunto de datos, selecciono el algoritmo y el filtro deseado podrá seleccionar el botón de **entrenar,** el cual se puede observar en la siguiente imagen.

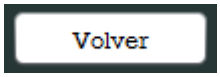
Una vez seleccionado el botón **entrenar**, el usuario debe esperar a que se entrene correctamente el algoritmo, cuando termine este proceso muestra la siguiente imagen.




### 2.4.2. Volver.

El Botón **Volver** permite al programa ir a la pantalla principal, la cual se muestra en la sección 1.3. PANTALLA INICIAL.

El botón se puede observar en la siguiente imagen.

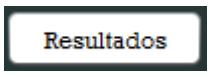



### 2.4.3. Resultados.

Este botón solo se habilita cuando que el algoritmo este entrenado.

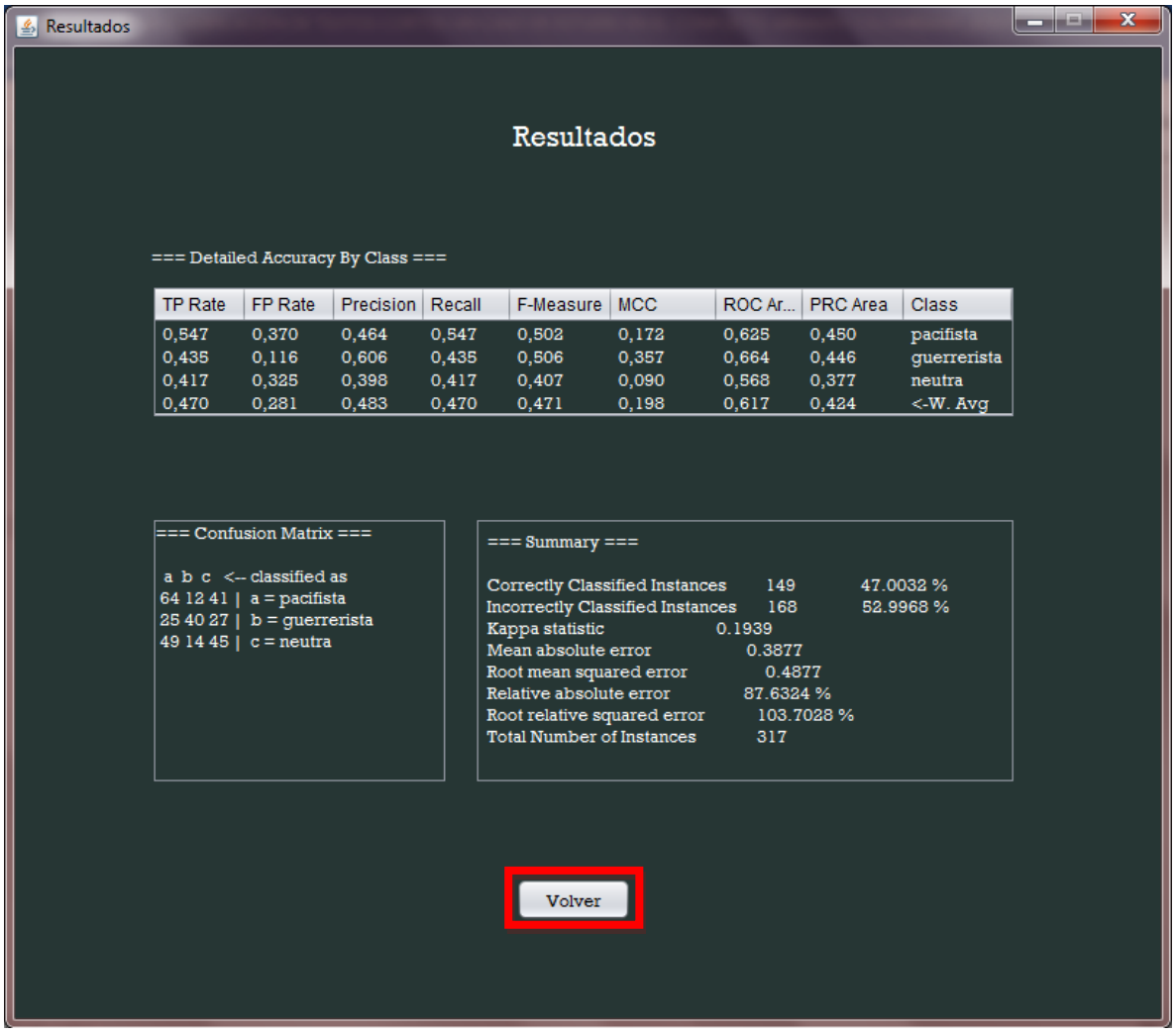
Al pulsar el botón **Resultados**, el programa redirige a la siguiente pantalla y a la siguiente sección de este documento.

El botón se puede observar en la siguiente imagen.



## 3. RESULTADOS

Cuando el algoritmo este entrenado y el usuario final seleccione el botón **Resultados**, se muestra la siguiente sección llamada Resultados, como se puede ver en la siguiente imagen.




El botón **Volver** permite al programa ir a la pantalla principal de la sección UTILIZACIÓN DEL SISTEMA la pantalla con nombre. **Carga y Análisis**

## ANEXO B: MANUAL DE PROGRAMADOR

En el presente anexo se presentan se muestran los elementos que componen el software desarrollado de una manera técnica.

UNIVERSIDAD CATÓLICA
de Colombia

# MANUAL DE PROGRAMADOR.


Presentado por:
**Juan Pablo Páramo Lozada. Código: 625381**
**Cesar Augusto Espitia Betancourt. Código: 625373**

Asesores:
Raúl Ernesto Menéndez Mora, PhD
Erika Paola Holguín Ontiveros.

Universidad Católica de Colombia
Facultad de ingeniera
Departamento de ingeniería de sistemas
Bogotá Colombia

2018

## TABLA DE CONTENIDO

## LISTA DE FIGURAS

# 1. INTRODUCCIÓN

El presente proyecto fue desarrollado en el lenguaje de programación Java y se utilizó la librería weka.jar lo que permitió agilizar el desarrollo del software lo que concuerda con la metodología planteada, a continuación, se mostrarán los elementos utilizados en el desarrollo del software con la finalidad de que el programa pueda ser modificado en futuros proyectos y mejorando la eficiencia de los algoritmos.
El software fue desarrollado en NetBeans IDE 8.2 con un JDK 1.8.0_191, también, aparte de la librería weka.jar se utilizó la librería Swing, está ya viene integrada en el JDK y es la que permite la realización de la interfaz gráfica.

## 1.1 Paquete Datos

Se crea un paquete especifico llamado “Datos” este paquete contiene las imágenes que se utilizan en las diferentes pantallas del software, además de esto contiene la base de datos entregada por la facultad de psicología con las modificaciones nombradas anteriormente, esta base de datos esta con extensión .arff y es llamada como “BD Tweets.arff”.

## 1.2 Paquete Algoritmos

El paquete “Algoritmos” contiene una clase llamada “Algoritmos”, esta clase contiene toda la lógica del software en dicha clase se va a realizar todo el procesamiento de los datos, es decir, se reciben los datos desde la interfaz, se guardan en una variable llamada “colección” en donde se guardan todas las instancias del set de datos y posteriormente se le asigna la clase en la cual debe clasificar, luego los datos son procesados, allí se encuentran todas las técnicas de pre procesamiento y los algoritmos, al utilizar la librería de weka basta solo con instanciar la técnica o el algoritmo, posteriormente se utiliza la función “buildclassiffier” pasándole como parámetro el set de datos almacenado en la variable “coleccion” luego se almacenan los resultados en 3 variables (strSummary, strDetailed, matriz), estas variables son declaradas de tipo static para poder ser utilizadas desde la interfaz para poder mostrar los resultados.

## 1.3 Paquete Interfaz

En este paquete se incluyen 3 clases, la primera clase llamada “WekaMain” es la clase principal, esta clase es la presentación del software en donde se muestra el título del proyecto, los autores, imágenes de la universidad católica y en la parte superior se encuentra un jmenú que permite ir a la siguiente pantalla, dicha pantalla está contenida en la clase “CargaAnalisis” allí se encuentran 2 grupos de botones, el primer grupo cuenta con 6 jradiobutton que permite elegir exclusivamente un botón al tiempo, cada uno de estos 6 jradiobutton

corresponden a cada uno de los algoritmos elegidos para que el usuario utilice, el segundo jgroup contiene 2 jradiobutton los cuales corresponden a 2 técnicas de pre procesamiento para ser aplicadas a cada algoritmo, al igual que el grupo anterior este solo permite elegir una técnica de pre procesamiento, en la parte inferior cuenta con 3 botones, el primer botón permite realizar el entrenamiento del algoritmo elegido, el segundo botón permite volver a la pantalla inicial, el tercer botón permite visualizar los resultados después de haber entrenado el algoritmo.

### 1.4 Archivo “BD Tweets.arff”

La base de datos está en estructura arff, se tiene la etiqueta @RELATION que permite asignarle un nombre a la base de datos para que pueda ser identificable por la plataforma, esto es debido a que se pueden cargar varias bases de datos en un mismo software utilizándolas para diferentes funciones, luego se encuentra la etiqueta @ ATTRIBUTE que permite declarar atributos, estos atributos pueden ser de 3 tipos: numeric, String o class, el atributo class contiene las clases en las que se debe clasificar, posterior a esto se encuentra la etiqueta @DATA la cual permite ubicar los datos.

# 2. CLASE ALGORITMOS

## 2.1 Datos

En esta clase se tienen algunas funciones relevantes, en la función “datos()” se utiliza la siguiente estructura:

```
DataSource ar = new DataSource(ruta);
coleccion = ar.getDataSet();
coleccion.setClassIndex(coleccion.numAttributes() - 1);
```

Se crea una variable de tipo DataSource donde se le pasa como parámetro la ruta lo que permite guardar todo el archivo cargado por el usuario, luego se declara una variable de tipo Instances llamada “coleccion” y esta va a contener el DataSet del archivo cargado, luego conociendo el flujo de datos de weka se le debe asignar una clase, esto se hace mediante la función “setClassIndex” y se le pasa como parámetro el último de los atributos declarados en el archivo.

## 2.2 Filtros

```
FilteredClassifier fclass = new FilteredClassifier();
StringToWordVector stwv=new StringToWordVector();
    fclass.setFilter(stwv);
    fclass.setClassifier(classifier);
```

Para colocar los filtros se declara una variable de tipo FilteredClassifier y se instancia el filtro que se desea aplicar al clasificador, esto se hace mediante la función “setFilter” y se le pasa como parámetro el filtro, luego la clase filtrada se le asigna el clasificador el cual es recibido como parámetro en la función.

```
fclass.buildClassifier(coleccion);
results(fclass);
```

Después la clase filtrada se utiliza para construir el clasificador mediante la función “buildClassiffier” pasándole como parámetro el set de datos y se llama la función resultados, esta función será explicada posteriormente, lo que hace es almacenar los resultados para luego ser enviados a la interfaz.

```
String [] options = Utils.splitOptions("-R first-last -W 1000
stwv.setOptions(options);
```

Existe otra manera de realizar los filtros de manera más específica, se declara un vector de opciones y en este se guardan las opciones del filtro, luego al filtro se le dice que utilice dichas opciones mediante la función “setOptions”, luego el proceso de filtro es igual al anterior.

### 2.3 Algoritmos

Para utilizar un algoritmo con esta librería simplemente basta con instanciarlo de la siguiente manera:

```
NaiveBayes nb=new NaiveBayes();
```

Este algoritmo se le va a pasar como parámetro al filtro elegido, para validar cual filtro elige el usuario la función recibe 2 parámetros los cuales van a ser los jradiobutton declarados dentro del jgroup2 en la interfaz y recibe el estado de cada jradiobutton, es decir, si esta seleccionado o no, luego mediante una validación se elige cual función de filtro enviar el algoritmo.

### 2.4 Resultados

Los resultados se ubican en la función “results()” allí se realiza la evaluación del rendimiento de cada algoritmo, para esto se declara una variable de tipo evaluación y se le pasan los datos ya filtrados, luego se le dice que realice una validación cruzada donde se le pasa el clasificador, los datos, el número de dobleces en el que va a dividir el set de datos y una semilla que permita garantizar la aleatoriedad, luego la variable eTest va a contener todos los resultados, para dividirlos se utilizan 3 variables (strSummary, strDetailed, matriz) estas permiten guardar los resultados de las funciones (toSummaryString(), toClassDetailsString(), toMatrixString()) respectivamente para luego ser pasados a la interfaz y ser mostrados en el software.

```
eTest = new Evaluation(coleccion);
eTest.crossValidateModel(classifier, coleccion, 10, new Random(1));
strSummary = eTest.toSummaryString();
strDetailed = eTest.toClassDetailsString();
matriz = eTest.toMatrixString();
```

## 3. CLASE WEKAMAIN

Esta es la clase principal por lo que es creada como un jFrame, la interfaz contiene un jmenubar que contiene un elemento jmenu el cual al darle clic dispara la acción de pasar a la pantalla de clasificación de algoritmos, la interfaz también cuenta con jlabels que permiten colocar las imágenes, el título del proyecto y los autores.

## 4. CLASE CARGAANALISIS

Esta clase es declarada como un jpanel, en la parte superior se tiene un jlabel para el título, abajo del título se tienen 2 elementos un jtextfield el cual va a obtener el nombre del archivo después de cargado y un botón que al ser presionado va a disparar una pantalla que permite encontrar el archivo en extensión arff dentro del equipo, y ser cargado en memoria, luego se observan 2 grupos de jradiobutton que como se mencionó anteriormente solo puede ser seleccionado tanto un algoritmo como una técnica, en la parte inferior se encuentran 3 botones, el primer botón “Entrenar” permite ejecutar la acción de la clase ubicada en algoritmos de lectura, modificación y evaluación de los datos, este le pasa el archivo cargado en memoria por el filechooser a dicha clase y la clase tiene listos los datos para que cuando el usuario de clic en el botón de resultados se cargue la clase resultados sin inconvenientes debido a que las variables ya tienen los resultados correspondientes.

## 5. CLASE RESULTADOS

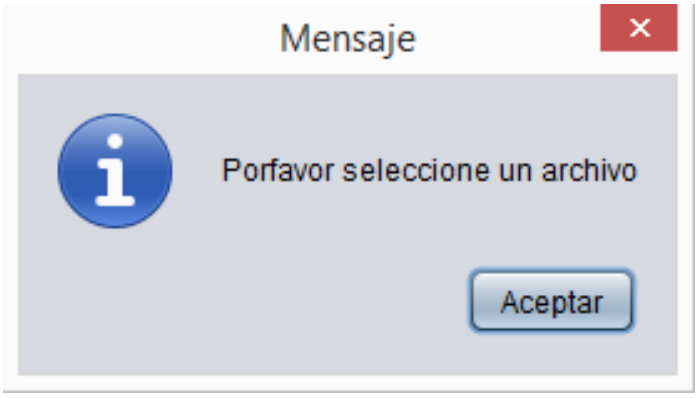
Esta clase, al no ser la principal se declara como un jpanel, cuenta con un jlabel para ubicar el título, cuenta con un jtable y 2 textarea, en un textarea se ubica la matriz, para esto se instancia la clase de “algoritmos” y posteriormente se trae la variable “matriz” y mediante un “setText” se muestran los datos en el textarea, el mismo proceso se realiza para el segundo textarea y se le envía como parámetro a la función “setText” la variable strSummary, para el contenido en la variable strDetailed se tiene un proceso diferente, al ser de tipo String se debe romper la cadena para ser ubicada en un vector y posteriormente ser ubicado en una matriz y con la matriz ya se puede llenar el jTable.

## 6. CHECK LIST

A continuación, se muestran las restricciones para evitar inconvenientes en el software:

Al darle clic en entrenar sin seleccionar un archivo mostrará un mensaje que le recuerde al usuario el cargue del mismo como se muestra en la siguiente figura:

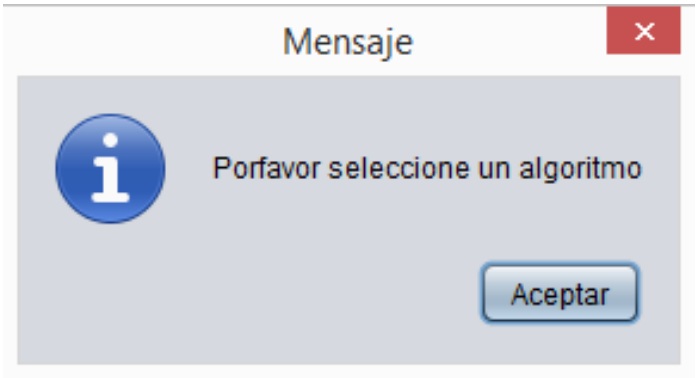
Figura 1. Mensaje de error de archivo.




Fuente: Autores.

Al darle clic en entrenar con un archivo seleccionado pero sin seleccionar un algoritmo se mostrará un mensaje que le recuerde al usuario seleccionar algún algoritmo como se muestra en la siguiente figura:

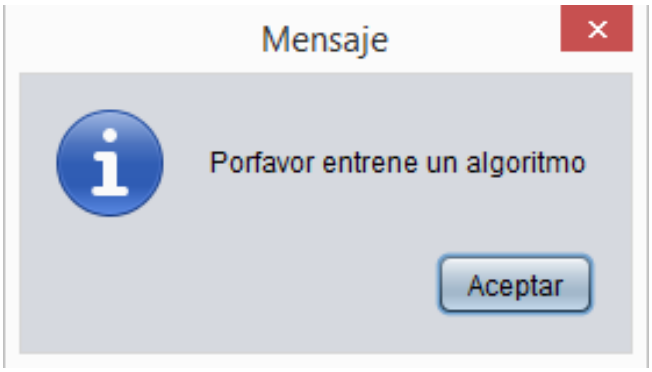
Figura 2. Mensaje de error de algoritmo.




Fuente: Autores.

Al darle clic en resultados sin haber entrenado un algoritmo se mostrará un mensaje que recuerde al usuario entrenar el algoritmo como se muestra en la siguiente figura:

Figura 3. Mensaje de error de entrenamiento.




Fuente: Autores.