Desarrollo de un componente de analítica para la clasificación de textos cortos dirigido a un proyecto institucional e integrable en una plataforma web

Juan Sebastián Jiménez Valero

Universidad Católica de Colombia
Facultad de Ingeniería
Programa de Ingeniería de Sistemas y Computación
Bogotá, Colombia
2018

Desarrollo de un componente de analítica para la clasificación de textos cortos dirigido a un proyecto institucional e integrable en una plataforma web

Juan Sebastián Jiménez Valero

Trabajo de grado presentado como requisito parcial para optar al título de:
Ingeniero de Sistemas

Asesor:
Ph.D., Raúl Ernesto Menéndez Mora

Alternativa:
Trabajo de auxiliar de investigación
Línea de Investigación:
Software inteligente y convergencia tecnológica.
Grupo de Investigación:
GISIC

Universidad Católica de Colombia
Facultad de Ingeniería
Programa de Ingeniería de Sistemas y Computación
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 Ingeniero de Sistemas.

_____

Jurado

_____

Jurado

_____

Raúl Ernesto Menéndez Mora, Ph. D
Asesor

A mi familia, especialmente a Santiago

Resumen

La clasificación de texto es una de las áreas de estudio de la disciplina del aprendizaje de máquina (en inglés *Machine Learning*) en donde se busca, posterior a una etapa de entrenamiento, predecir una categoría para datos de entrada que no hayan sido clasificados previamente.

La longitud de los textos cortos, puede conllevar a una pérdida en la precisión de los resultados entregados por el proceso de clasificación de texto, ya que la cantidad de características aprovechables disminuye. Por lo tanto, se busca explorar una solución que permita realizar tareas de clasificación de textos cortos, con un nivel de precisión cercano al 80 %.

Se desarrolló un componente de clasificación de textos cortos en el lenguaje de programación Python, haciendo uso del framework Flask el cual permite peticiones a través de un API y realiza la clasificación datasets que cumplan con el formato de entrada. Se probaron los resultados de este trabajo mediante el uso de publicaciones extraídas desde cuentas de Twitter, debido a la restricción sobre la longitud de sus publicaciones. La clasificación se realizó mediante el uso de algoritmos de aprendizaje supervisado, y en el mejor de los casos, la precisión obtenida fue cercana al 85 %.

Palabras clave: Análisis automático de textos, inteligencia artificial, reconocimiento de patrones.

Abstract

The text classification is an area of study of Machine Learning discipline in which the objective, after a training process, is predict the resultant category of unlabeled input data.

The lenght of short texts, could carry to a leak of accuracy in the text classification task, since the amount of usable characteristics decreases. Accordingly, is desired to find a solution that could help on short text classification tasks, with an accuracy's level near to 80 %.

A short text classification component was developed in the Python programming language, making use of the Flask framework which allows requests through an API and performs the classification of datasets that comply with the input format. The results of this work were tested through the use of publications extracted from Twitter accounts, due to the restriction on the length of their publications. The classification was made through the use of supervised learning algorithms, and in the best of cases, the accuracy obtained was close to 85 %

Keywords: Artificial intelligence, automatic text analysis, pattern recognition

Contenido

Lista de Figuras

Lista de Tablas

1. INTRODUCIÓN

Psicológicamente hablando, se pueden categorizar a las personas según su comportamiento y su discurso. El macro proyecto universitario *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"* propone realizar una categorización psicológica de los diferentes actores participantes en el conflicto armado colombiano.

Los procesos de clasificación de texto, constituyen una de las áreas de estudio del aprendizaje automático. El aprendizaje de máquina *(Machine Learning)* es una disciplina de la inteligencia artificial encargada del diseño de algoritmos que permiten a las computadoras encontrar patrones a partir de un conjunto de datos y lograr aprender de los mismos[1].

El trabajo realizado con estos datos, requiere que sean gestionados y agrupados en diferentes categorías; esta agrupación nos ayuda a comprender ciertas características de los fenómenos que sean objeto de estudio para posteriormente realizar comparaciones con fenómenos relacionados que hayan sido previamente estudiados[2]. En este tipo de algoritmos, se entrena al sistema para aprender cómo debe realizar la clasificación de los datos de entrada según una serie de datasets[3] de entrenamiento.

A partir de las categorías definidas en el proyecto institucional *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"* que son asignables a los discursos expuestos por los actores del conflicto y que describen rasgos generales de diferentes categorías psicológicas, pasamos a preguntarnos ¿Cómo se podrían asignar categorías psicológicas a los discursos expuestos en redes sociales por los diferentes actores del conflicto de manera automática, a fin de analizarlos más profundamente? ¿Cuál es el proceso que se debe llevar a cabo para tal fin? ¿Ayudarán las técnicas de aprendizaje automático a realizar esta tarea de clasificación?.

[1] Tom M Mitchell. *Machine Learning*. Vol. 4. 1997, págs. 417-433. ISBN: 9781577354260. DOI: 10.1145/242224.242229. arXiv: 0-387-31073-8, William L. (Encyclopædia Britannica) Hosch. *machine learning*. 2009. URL: https://global.britannica.com/technology/machine-learning (visitado 26-03-2017).

[2] Rui Xu. "Survey of clustering algorithms for MANET". En: *IEEE Transactions on Neural Networks* 16.3 (2005), págs. 645-678. ISSN: 1045-9227. DOI: 10.1109/TNN.2005.845141. arXiv: 0912.2303. URL: http://arxiv.org/abs/0912.2303.

[3] El término dataset hace referencia a un conjunto de datos

En este documento se busca establecer una propuesta formal que permita abordar la problemática de clasificación de textos cortos, incluyendo en dicha propuesta la utilización de algoritmos destinados a tareas de clasificación, planteados en el aprendizaje automático. Se espera entregar como resultado del proceso las diferentes categorías a las cuales pertenecen los datos del conjunto de entrada.

Como caso de estudio se limitarán los datos de entrada a los resultados provenientes del proceso de extracción realizado sobre la red social *Twitter* por la plataforma propuesta como solución para el desarrollo del macro proyecto *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"*.

1.1. PLANTEAMIENTO DEL PROBLEMA

El conflicto armado interno en Colombia, se remonta desde mediados del siglo XX, donde surgen grupos militantes que se alzan en contra del Estado colombiano[4], provocando un aproximado de 218.000 muertos, cerca de 5.700.000 desplazamientos forzados, 25.000 desaparecidos, entre otras cifras resultantes de dicho conflicto, contabilizado desde el año de 1958 y hasta el año 2013[5].

La participación en el conflicto armado puede generar diferentes traumatismos los cuales tienen repercusión directa en las emociones de las personas. Para minimizar el impacto de dicha participación, los diferentes grupos buscan legitimar mediante justificaciones ideológicas la perpetración de acciones violentas, basados en discursos que reconocen la existencia del conflicto y que acentúan la diferenciación grupal.

En este contexto, las redes sociales son usadas para la expresión de ideas, legitimadoras o no, a través de la exposición a estos discursos. En este estudio, mediante la clasificación de información extraída de la red social *Twitter*, se buscará determinar si esta pertenece a orientaciones dadas a la paz o a la guerra.

En el marco de este trabajo investigativo se requieren datos provenientes de un proceso de extracción y almacenamiento. El análisis estadístico y la visualización de dichos datos recolectados de las redes sociales de personajes participantes en el conflicto son insumos básicos,

[4] Valentina Jaramillo Bustamante. "Conflicto armado en Colombia, el proceso de paz y la Corte Penal Internacional: Un estudio sobre la internacionalización del conflicto armado en Colombia y su búsqueda por encontrar la paz duradera". En: *Journal of International Law* 6.6 (2015), págs. 6-34. ISSN: 2216-0965.

[5] Centro Nacional de Memoria Histórica. *Estadísticas - ¡Basta ya! Colombia: Memorias de guerra y dignidad.* 2016. URL: `http://www.centrodememoriahistorica.gov.co/micrositios/informeGeneral/estadisticas.html` (visitado 28-03-2017).

para poder analizar las diferentes categorías en las que se puedan clasificar estos discursos emitidos por actores del conflicto armado.

Partiendo de lo anterior se desprende la siguiente incógnita: ¿Cómo se puede implementar un componente de analítica que realice una clasificación automática de textos cortos y que sea integrable a la plataforma propuesta para el proyecto institucional *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"*?

1.2. JUSTIFICACIÓN

La clasificación corresponde a una serie de tareas, que buscan dividir objetos en ciertas categorías que los describan según las características que se estén evaluando en el momento. A su vez, la clasificación de textos, corresponde a una serie de algoritmos de aprendizaje supervisado, propios de la disciplina de machine learning; los cuales tienen como objetivo la categorización de datos, según la similitud con ciertas características grupales, obtenidas a través del estudio de una serie de datos de entrenamiento, los cuales se encuentran previamente categorizados.

Los datos de entrenamiento deben ser clasificados mediante procesos manuales que se realizan por expertos en él área del conocimiento que se esté estudiando, esto hace que el proceso sea más lento que las posibles soluciones computarizadas, por lo tanto, de hacer todo el proceso de manera manual extendería significativamente el tiempo que toma realizar la clasificación.

La asignación de categorías, se hace desde un conjunto de valores discretos, que contiene cada una de las categorías disponibles para ser asignadas a los datos de entrada, y el dataset de entrenamiento sirve para la generación de modelos de clasificación los cuales amplían las características que se deben cumplir para pertenecer a determinada categoría[6].

Sin embargo, la longitud de los textos cortos puede conllevar a la disminución de la precisión en las clasificaciones realizadas ya que la cantidad de información aprovechable disminuye, es por eso que se requiere de una etapa de entrenamiento que sea rigurosa, buscando aumentar la precisión entregada por los algoritmos, ya que al clasificar los datos de entrada se dará como resultado un porcentaje de certidumbre en la misma, con esto se busca que dicho porcentaje sea similar al ofrecido por los expertos.

[6] Charu C. Aggarwal y Chengxiang Zhai. *Mining Text Data*. Vol. 8. New York: Springer Science & Business Media, 2012, pág. 524. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-4614-3223-4, Capítulo 6.

Por lo tanto, se busca constatar la aplicabilidad del concepto de clasificación de texto, a la información extraída desde redes sociales, para clasificar de manera automática el discurso. De esta forma se lograría, una disminución en los tiempos que toma el análisis de los datos y su clasificación en diferentes categorías. Lograr esta tarea en tiempo real permitiría hacer un análisis en linea de las tendencias discursivas de los actores del conflicto.

Teniendo en cuenta que el proyecto *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"* cuenta con componentes de extracción de datos y análisis estadístico, estos podrían ser usados como insumo para el diseño de una solución al problema de clasificación, dado que estos brindan información almacenada de manera coherente (componente de extracción), y un primer análisis de la misma (componente de análisis estadístico), los cuales facilitan el trabajo de clasificación de los datos.

Los resultados de esta propuesta se verán reflejados en una disminución en los tiempos requeridos para clasificar los discursos en diferentes categorías, así como en la automatización de este procedimiento, ayudando a definir estrategias para la reexperimentación emocional y mindfulness.

1.3. OBJETIVOS

1.3.1. Objetivo general

Desarrollar un componente de software para la clasificación de textos cortos, que se pueda integrar en la plataforma propuesta como solución para las diferentes necesidades del macro proyecto: *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"*.

1.3.2. Objetivos específicos

1. Caracterizar el estado del arte de los algoritmos de clasificación de texto más usados y su aplicación en problemas similares.

2. Diseñar la arquitectura definida para el desarrollo del componente de software para clasificación, teniendo en cuenta la necesidad de la aplicación de ingeniería de los rasgos, parametrización y presentación de resultados.

3. Implementar un componente de software que se integre con la plataforma propuesta para la extracción, manipulación y visualización asociada al proyecto institucional *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos*

expuestos a situaciones traumáticas en un contexto de violencia política", y que permita realizar la tarea de clasificación propuesta en éste, involucrando en su desarrollo al menos cinco algoritmos de clasificación de texto.

4. Aplicar una serie de pruebas que permitan garantizar el correcto funcionamiento del componente de software.

1.4. DELIMITACIÓN

1.4.1. Espacio geográfico

- El trabajo de grado se realiza en la Universidad Católica de Colombia ubicada en la ciudad de Bogotá, Colombia.

1.4.2. Tiempo

- El tiempo de desarrollo e implementación estarán limitados al tiempo que dure un período académico.

1.4.3. Contenido

- Este documento contiene la información consultada que se consideró relevante durante el planteamiento del desarrollo de un componente de software para la clasificación de textos cortos, incluyendo el estado del arte.

1.4.4. Alcance

- Se deberá entregar un componente de software, que permita realizar tareas de clasificación de textos cortos en determinadas categorías.

- El componente deberá permitir su integración con el framework propuesto para abordar la solución a la problemática planteada en el proyecto *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"*.

- Como insumo para las pruebas del componente de software, se utilizarán textos cortos extraídos de la red social *Twitter* como caso de estudio. Estos textos serán provistos por el usuario.

1.4.5. Limitaciones

- La solución planteada por el macroproyecto *"Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política"* conlleva a un proceso de desarrollo por fases, dentro de estas el producto de software busca cumplir la función de una fase intermedia, la cual consiste en la clasificación de los textos dados como insumo en categorías previamente definidas.
- El desarrollo, la implementación y la documentación del producto de software son tareas ejecutadas por una sola persona.
- Debido a la duración del proyecto una vez alcanzada la fecha de cierre y terminado el desarrollo, el componente entregado no contará con mantenimiento o modificaciones adicionales por parte del desarrollador.
- La infraestructura disponible para los procesos involucrados con el desarrollo del producto es básica y limitada.

2. MARCO DE REFERENCIA

2.1. MARCO CONCEPTUAL

- **Bigramas:** El término hace referencia a la agrupación de dos unidades escritas consecutivas, en este caso palabras[1].

- **Blog:** Es un sitio web que se actualiza constantemente y en donde el autor puede publicar información que sea de su interés, generalmente las publicaciones del sitio se organizan en forma cronológica inversa, es decir primero se mostrarán las publicaciones o entradas más recientes.[2]

- **Datos Abiertos:** "Son todos aquellos datos primarios o sin procesar, que se encuentran en formatos estándar e interoperables que facilitan su acceso y reutilización, los cuales están bajo la custodia de las entidades públicas o privadas que cumplen con funciones públicas y que son puestos a disposición de cualquier ciudadano, de forma libre y sin restricciones, con el fin de que terceros puedan reutilizarlos y crear servicios derivados de los mismos."[3]

- **Flask:** Es un framework para el desarrollo de aplicaciones web en python, se definen como un micro framework ya que integran extensiones a su base para proveer de funcionalidades a los programas que soportan.[4]

- **Framework:** Se puede definir en este contexto como un compendio de reglas usadas para la creación o planificación de algo.[5]

---

[1] Oxford University Press. *Definition of bigram*. 2018. URL: https://en.oxforddictionaries.com/definition/bigram (visitado 14-04-2018).

[2] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). *¿Qué es un blog?* URL: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo%7B%5C_%7D1%7B%5C_%7DIniciacionblog/qu%7B%5C_%7Des%7B%5C_%7Dun%7B%5C_%7Dblog.html (visitado 11-06-2017).

[3] "LEY 1712 DE 2014". En: *Diario Oficial 49084 de marzo 6 de 2014* (2014). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=56882.

[4] Armin Ronacher. *Foreword — Flask Documentation (0.12)*. 2010. URL: http://flask.pocoo.org/docs/0.12/foreword/ (visitado 15-04-2018).

[5] Cambridge University Press. *Definition of framework*. 2018. URL: https://dictionary.cambridge.org/es/diccionario/ingles/framework (visitado 15-04-2018).

- **Gamificación:** Se refiere al uso de conceptos propios al desarrollo de juegos en contextos que no están relacionados con el juego, como por ejemplo tableros de puntuación o recompensas.[6]

- **Historias de usuario:** "Las historias de usuario son la técnica utilizada en *extreme programming* para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas."[7]

- **Lematización:** Es el proceso mediante el cual se buscan las raíces comunes de una palabra.[8]

- **Microblogging:** Es una variante a los blogs que se carácteriza porque sus entradas son generalmente más cortas.[9]

- **Pruebas de aceptación:** Buscan aprobar la correcta funcionalidad de las historias de usuario mediante una serie de pruebas, en donde con base a una entrada se verifica si la salida obtenida corresponde con la esperada, este tipo de pruebas se debe validar en conjunto con el cliente.[10]

- **Pruebas unitarias:** Son pruebas que se hacen al código descomponiéndolo en sus unidades funciones más básicas, una vez se hayan superado las pruebas unitarias más granulares, se deberán tomar más partes de código y realizar nuevas pruebas, para asegurar el funcionamiento correcto de todo el sistema de software.[11]

[6] Giorgio Maria Di Nunzio, Maria Maistro y Federica Vezzani. "A Gamified Approach to NaiVe Bayes Classification: A Case Study for Newswires and Systematic Medical Reviews". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1139-1146. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191547. URL: https://doi.org/10.1145/3184558.3191547.

[7] Patricio Letelier y Mª Carmen Penadés. "Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". En: (). URL: http://users.dsic.upv.es/asignaturas/eui/lds/doc/masyxp.pdf.

[8] Real Academia Española. *DLE: lematizar*. 2014. URL: http://dle.rae.es/?id=N6LviQx (visitado 15-04-2018).

[9] Julián Pérez Porto y Ana Gardey. *Definición de Twitter*. 2014. URL: http://definicion.de/twitter/ (visitado 11-06-2017).

[10] José Joskowicz. "Reglas y Prácticas en eXtreme Programming". En: (2008). URL: https://iie.fing.edu.uy/%7B~%7Djosej/docs/XP%20-%20Jose%20Joskowicz.pdf, Álvaro Galván Lucas y Jose Manuel Torres Púa. *Extreme programming*. URL: http://osl2.uca.es/wikiCE/index.php/Extreme%7B%5C_%7Dprogramming (visitado 09-08-2017).

[11] Álvaro Galván Lucas y Jose Manuel Torres Púa. *Extreme programming*. URL: http://osl2.uca.es/wikiCE/index.php/Extreme%7B%5C_%7Dprogramming (visitado 09-08-2017).

- **Texto corto:** Se considera un texto corto a uno cuya extensión se encuentre entre los 30 y los 400 caracteres, generalmente son utilizados en publicaciones de redes sociales, mensajes de chat, o mensajes de texto; Se caracterizan por su carácter de comunicación inmediata por lo que cuentan con pocas palabras y características aprovechables, en muchas ocasiones se usan términos del argot.[12]

- **Trigramas:** El término hace referencia a la agrupación de tres unidades escritas consecutivas, en este caso palabras[13].

- **Tuit:** El término Tuit o en inglés *Tweet* son todas aquellas publicaciones que pueden enviar los usuarios registrados en Twitter, las cuales constan de un máximo de 280 caracteres, los tuits pueden incluír imágenes, vídeos o enlaces a sitios web. De forma predeterminada estos son públicos.[14]

- **Twitter:** Twitter es una red social de *microblogging* originada en el año 2006 que permite realizar actualizaciones en tiempo real, se caracteriza por la longitud de sus entradas, más conocidas como tuits.[15]

- **Wordnet:** Es una base de datos orientada al almacenamiento de términos léxicos del idioma inglés; los términos contenidos en esta son agrupados por relaciones semánticas y conceptuales.[16]

[12] Ge Song y col. "Short Text Classification: A Survey". En: *Journal of Multimedia* 9.5 (2014), págs. 635-643. ISSN: 1796-2048. DOI: 10.4304/jmm.9.5.635-643. URL: http://ojs.academypublisher.com/index.php/jmm/article/view/12635.

[13] Oxford University Press. *Definition of trigram*. 2018. URL: https://en.oxforddictionaries.com/definition/trigram (visitado 14-04-2018).

[14] Twitter International Company. *Política de Privacidad de Twitter*. 2017. URL: https://twitter.com/privacy?lang=es%7B%5C#%7Dupdate (visitado 11-06-2017).

[15] Julián Pérez Porto y Ana Gardey. *Definición de Twitter*. 2014. URL: http://definicion.de/twitter/ (visitado 11-06-2017).

[16] Christiane Fellbaum. *WordNet — A Lexical Database for English*. 2005. URL: https://wordnet.princeton.edu/ (visitado 15-04-2018).

2.2. MARCO TEÓRICO

2.2.1. Machine learning

Según Mitchell[17] la disciplina del aprendizaje de máquina o *Machine Learning* busca estudiar la construcción de sistemas, que vayan mejorando según la experiencia, adquirida mediante el tratamiento del problema que se está estudiando. Se puede decir, que un sistema aprende de una experiencia **E**, si su desempeño en la realización de la tarea **T**, con base a la medida de desempeño **P** muestra mejoría a través de varias iteraciones resolviendo el problema.

Para ejemplificar esto, Gonzalez[18] propone la revisión del siguiente caso de estudio:

Si tenemos un algoritmo cuyo objetivo es jugar a las damas, la manera en la cual podríamos relacionarlo con el esquema mencionado, sería así:

- Se dice que la tarea **T** es jugar a las damas.
- La medida de desempeño **P** será el porcentaje de juegos ganados contra oponentes.
- La experiencia de entrenamiento **E** por su parte será jugar partidas de práctica contra si mismo

Por lo tanto diremos que el algoritmo aprende si a través de las iteraciones el porcentaje de juegos ganados va en aumento.

El aprendizaje de máquina, busca ayudar a solucionar problemas que serían prácticamente intratables para el ser humano, ya sea por la complejidad de los mismos, o la cantidad de información que se debe tener en cuenta para la solución de estos.

Según lo mencionado por Brownlee[19] se pueden clasificar los tipos de aprendizaje de máquina como:

Aprendizaje supervisado

Cuando se parte de un conjunto de datos del cual se conoce la salida Y_n correspondiente a la entrada X_n, se busca generalizar un modelo que permita predecir para datos de entrada

[17] Tom M Mitchell. "The Discipline of Machine Learning". En: *Machine Learning* 17.July (2006), págs. 1-7. ISSN: 0264-0414. DOI: 10.1080/026404199365326. arXiv: 9605103 [cs]. URL: http://www-cgi.cs.cmu.edu/%7B~%7Dtom/pubs/MachineLearningTR.pdf.

[18] Fabio A. González. *Introducción Aprendizaje de Máquina*. 2007. URL: http://dis.unal.edu.co/profesores/fgonza/courses/2007-I/ml/ml-01-introduction.pdf (visitado 11-06-2017).

[19] Jason Brownlee. *Supervised and Unsupervised Machine Learning Algorithms*. 2016. URL: http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (visitado 06-06-2017).

nuevos su correspondiente salida.

Se conoce como aprendizaje supervisado ya que con los datos de entrenamiento, el sistema trata de realizar predicciones sobre la posible salida. Estas predicciones son corregidas hasta que se alcanza un nivel de precisión aceptable.

Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, en este modelo de aprendizaje solo se cuenta con el dato de entrada X_n, y ninguna salida que corresponda, en este caso se busca encontrar distribuciones en los datos de entrada, o patrones relacionados a los mismos.

Aprendizaje semi-supervisado

En este caso se cuenta con un conjunto de datos de entrada de gran tamaño, en donde solo para una pequeña fracción de los mismos se conoce la salida correspondiente. En esta categoría se encuentra la mayoría de los problemas atacados por el aprendizaje de máquina, debido a que puede resultar costoso etiquetar cada dato de entrada, con su respectiva salida.

Aprendizaje por refuerzo

Este tipo de aprendizaje busca que la máquina aprenda a través de la experiencia que obtiene mediante la realización de diferentes iteraciones de solución de un problema, en este caso no se usan datos de entrenamiento sino que las decisiones que tome el algoritmo vendrán con una recompensa o castigo según sea el caso, aprendiendo de los resultados obtenidos sobre las decisiones tomadas se busca maximizar el valor obtenido a largo plazo[20].

2.2.2. Clasificación de textos

La clasificación de textos, constituye una categoría de algoritmos supervisados, donde la variable de salida es conocida. Según Aggarwal[21] éstos algoritmos buscan agrupar una cantidad finita de datos en un conjunto también finito de grupos o categorías. En este caso se entrena al sistema para aprender cómo debe realizar la clasificación de los datos de entrada según una serie de *datasets* de entrenamiento, los cuales están previamente clasificados por un experto, de allí se genera un modelo de clasificación que relaciona las características del dato que se está analizando con una de las categorías de destino.

[20] Vishal Maini y Samir Sabri. *Reinforcement Learning*. 2017. URL: https://medium.com/machine-learning-for-humans/reinforcement-learning-6eacf258b265 (visitado 14-04-2018).

[21] Charu C. Aggarwal y Chengxiang Zhai. *Mining Text Data*. Vol. 8. New York: Springer Science & Business Media, 2012, pág. 524. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-4614-3223-4, Capítulo 6.

El problema de la clasificación de textos se puede generalizar en dos grandes ramas, que abarcan el problema; cuando solo se debe asignar una categoría al dato de entrada se considera que la solución es más compleja de hallar o cuando probabilísticamente se asigna el dato a varias categorías lo cual resulta menos complejo.

Las aplicaciones para el concepto de clasificación de textos se extienden por diversos campos de influencia, como las recomendaciones de productos, filtros anti spam de los correos electrónicos o recuperación de información.

En el contexto del proyecto de investigación se buscará clasificar los datos de entrada en una única categoría.

2.2.3. Clasificación de textos cortos

Dentro del problema estudiado por la clasificación de textos, se presenta un subconjunto para su estudio, el cual trata la clasificación de textos con una cantidad reducida de palabras, como lo pueden ser las publicaciones de los usuarios en redes sociales; este problema es importante ya que de esta información se pueden extraer opiniones del usuario e inferir tendencias basadas en el comportamiento del mismo, esta última aplicación es ampliamente utilizada para la publicidad segmentada de acuerdo a preferencias de usuario.

En este subconjunto del problema de clasificación se debe hacer frente a ciertas dificultades clave para el proceso las cuales son la longitud de las públicaciones extraídas, y la cantidad de publicaciones que puedan estar relacionadas entre sí, esto conlleva a que antes de realizar cualquier tipo de análisis sobre la información se deba aplicar un proceso de limpieza de la misma y pre clasificación donde se separen los datos que puedan ser útiles de los que no lo sean, lo cual representa un aumento en el tiempo previo al proceso de clasificación que se realiza sobre información considerada útil, esto incrementa la dificultad para realizar análisis en tiempo real[22].

2.2.4. Ingeniería de rasgos

Los datos son parte fundamental en todo el proceso realizado por el aprendizaje de máquina, por lo tanto al trabajar con estos modelos predictivos se busca máximizar la información útil para el proceso, y así aumentar la calidad de los resultados. Por ende, esto debe ir soportado por la extracción correcta de las características a analizar, según el problema que se esté enfrentando, ya que existen características irrelevantes para el proceso, las cuales pueden

[22]Faris Kateb y Jugal Kalita. "Classifying Short Text in Social Media: Twitter as Case Study". En: *International Journal of Computer Applications* 111.9 (2015), págs. 1-12. ISSN: 09758887. DOI: 10.5120/19563-1321. URL: http://research.ijcaonline.org/volume111/number9/pxc3901321.pdf.

llegar a entorpecerlo[23].

Según Brownlee[24] la ingeniería de rasgos busca solucionar el problema de obtener la mayor cantidad de características aprovechables a partir de los datos de entrada, ya que esto influye directamente en el éxito del módelo predictivo usado en la solución de la problemática estudiada. Otro resultado de una buena selección de características es la flexibilidad que ofrecen al momento de ejecutar los modelos predictivos, ya que estos pueden entregar resultados buenos utilizando algoritmos que sean menos complejos.

2.2.5. Revisión de algunos algoritmos de clasificación

El estudio de la información contenida en textos cortos se remonta al uso de los SMS *Short Message Service* hace aproximadamente 20 años. La popularización del uso de las redes sociales ha incrementado sustancialmente la cantidad de información disponible, y esto, a su vez genera la necesidad en la comunidad científica de desarrollar técnicas que permitan enfrentar los retos planteados por el estudio de textos cortos, como lo son la cantidad de información y la longitud de los datos que son objeto de estudio, por lo tanto los autores Kateb y Kalita[25] proponen afrontar el problema con el uso de técnicas aplicadas en otras áreas del conocimiento, que se puedan ajustar a la clasificación de textos cortos.

Por su parte para la clasificación de textos Aggarwal[26] agrupó y definió las técnicas más utilizadas como:

- **Árboles de decisión:** Un árbol de decisión es una descomposición jerárquica que se realiza sobre el *dataset* de pruebas, en donde se busca que un predicado o una condición dividan los datos. En el ámbito de la clasificación de textos, estos predicados atienden a la presencia o ausencia de ciertas palabras en el texto, lo cual dota de sentido al mismo.

  En este tipo de clasificadores luego de generar una estructura arbórea, se utilizarán datos que son parte del conjunto de entrenamiento, y que no fueron usados para la

[23] Charu C. Aggarwal y Chengxiang Zhai. *Mining Text Data*. Vol. 8. New York: Springer Science & Business Media, 2012, pág. 524. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-4614-3223-4, Capítulo 6.

[24] Jason Brownlee. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. 2014. URL: http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (visitado 03-06-2017).

[25] Faris Kateb y Jugal Kalita. "Classifying Short Text in Social Media: Twitter as Case Study". En: *International Journal of Computer Applications* 111.9 (2015), págs. 1-12. ISSN: 09758887. DOI: 10.5120/19563-1321. URL: http://research.ijcaonline.org/volume111/number9/pxc3901321.pdf.

[26] Charu C. Aggarwal y Chengxiang Zhai. *Mining Text Data*. Vol. 8. New York: Springer Science & Business Media, 2012, pág. 524. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-4614-3223-4, Capítulo 6.

generación de la estructura, para verificar si esta es lo suficientemente granular para solucionar el problema; ya que se pondrán a prueba los datos y de poder segmentarlos aún más en nuevas subcategorías (*hojas*) se deberá dividir nuevamente la rama del árbol para lograr que la clasificación sea lo más detallada posible.

- **Clasificadores basados en reglas:** Estos clasificadores establecen un concepto similar a los árboles de decisión ya que estos, se rigen por un conjunto de reglas que condicionan la pertenencia a una clase esperada. Por lo general, estas reglas validan la existencia de ciertos predicados dentro del texto, más no su ausencia como lo hacen los árboles de decisión; sin embargo, toda una rama de un árbol se puede expresar como una regla para un clasificador de este tipo.

  Generalmente las reglas son condiciones sencillas y representan un conjunto de términos que deben estar presentes en el texto para cumplir con la condición planteada de pertenencia a una clase. El objetivo principal de este tipo de clasificadores es generar un conjunto de reglas que cubra todos los posibles predicados que puedan estar en el texto a clasificar al menos por una regla.

- **Máquinas de soporte vectorial:** Estas basan su funcionamiento en la determinación de diferencias entre las clases que se tienen disponibles para la clasificación de los datos de entrada. Estas diferencias trazan límites de pertenencia a las clases, ya que los textos de una misma clase guardan cierta relación entre si, lo cual brinda soporte a las tareas de clasificación en donde se determinará en base a estas diferencias si el dato pertenece o no a determinada clase.

- **Redes Neuronales:** Las redes neuronales se relacionan con las máquinas de soporte vectorial ya que ambos modelos buscan realizar las tareas de clasificación a partir de las diferencias que separan una clase de la otra, para este caso las redes neuronales artificiales (ANN, del inglés *Artificial Neural Network*), se compone de varias unidades o "neuronas", las cuales a partir de un conjunto de datos de entrada, brindan una salida enmarcada dentro de estas diferencias. La clasificación que se realiza dentro de una red neuronal se basa en el peso que tengan las palabras contenidas en el texto estudiado. Los pesos en la red son modificados con el fin de reducir el margen de error en la tarea realizada, en este tipo de clasificadores se suelen usar varias capas de neuronas que son alimentadas por el proceso realizado en las capas inferiores[27]; el objetivo de tener varias capas es mejorar la precisión en la selección de la clase de salida mediante el análisis de varios límites de la misma.

[27] Tania Camila Niño y col. "Uso de redes neuronales artificiales en predicción de morfología mandibular a través de variables craneomaxilares en una vista posteroanterior". En: (2016). DOI: http://dx.doi.org/10.11144/Javeriana.uo35-74.urna.

- **Clasificadores Bayesianos:** Dentro de los clasificadores probabilísticos se encuentran los clasificadores ingenuos (del inglés, *naive*) de Bayes; estos son ampliamente usados a pesar de su simplicidad, ya que su base teórica entrega resultados sobresalientes para varias tareas de clasificación.

  A estos clasificadores se les llama ingenuos porque modelan la distribución de los términos presentes en los textos trabajando bajo la presunción de que la distribución de diferentes términos (es decir su presencia o ausencia) es independiente de la de otros. Esto mejora la calidad de los resultados obtenidos al trabajar en ambientes con variables "ruidosas" como la clasificación de textos, donde no todas las palabras contenidas en el texto a estudiar contienen significado, o aportan a la tarea de clasificación[28].

[28] Irina Rish. "IBM Research Report An empirical study of the naive Bayes classifier". En: (2001). URL: `http://domino.watson.ibm.com/library/CyberDig.nsf/home`.

2.3. MARCO LEGAL

2.3.1. Normatividad colombiana

Debido a que el país carece de normatividad que trate directamente sobre la información que se encuentre en redes sociales, se debe trabajar bajo el marco referencial de los derechos a la privacidad de los ciudadanos, como se consagra en la Constitución Política de Colombia; la cual en el artículo 15 establece el derecho a la intimidad y la responsabilidad que tiene el Estado colombiano en el respeto a este derecho.

> "Todas las personas tienen derecho a su intimidad personal y familiar y a su buen nombre, y el Estado debe respetarlos y hacerlos respetar. De igual modo, tienen derecho a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos y en archivos de entidades públicas y privadas.
>
> En la recolección, tratamiento y circulación de datos se respetarán la libertad y demás garantías consagradas en la Constitución..."[29]

Este artículo fundamenta la ley de *Habeas Data* que tuvo origen en el año 2012 mediante la Ley estatutaria 1581, mediante la cual se reglamenta el tratamiento, la confidencialidad y el uso que se le dé a los datos recolectados. De igual forma la ley en cuestión compromete a la persona o entidad que esté recolectando, tratando o distribuyendo estos datos a informarle al dueño de dicha información la política de tratamiento que le aplicaría a esta según los objetivos para los que se esté recolectando[30].

También mediante el Decreto 1377 de 2013 se reafirma la autonomía que tienen las personas o entidades sobre los datos que se recolectan de ellas y el derecho que tienen a solicitar la consulta, modificación o supresión de esta información, para lo cual la persona o entidad responsable de los datos debe tener medios accesibles y gratuitos al público[31].

En el marco del proyecto de investigación y teniendo en cuenta que se analizará, entre otra, información proveniente de personas y entidades públicas, se aplicaría también el contexto de la ley de acceso a la información pública o *datos abiertos*, esta se reglamenta mediante la Ley 1712 de 2014, en donde se dispone que la recolección y tratamiento de datos se realizará

[29] República de Colombia. Corte Constitucional. "Constitución Política de Colombia". En: Normatividad 5 - 2015 (1991), pág. 125. URL: http://www.corteconstitucional.gov.co/inicio/Constitucion%20politica%20de%20Colombia%20-%202015.pdf, artículo 15.

[30] "LEY ESTATUTARIA 1581 DE 2012". En: *Diario Oficial 48587 de octubre 18 de 2012.* (2012). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=49981.

[31] "DECRETO 1377 DE 2013". En: *Diario Oficial 48834 del 27 de junio de 2013* (2013). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=53646%7B%5C#%7D0.

siguiendo principios éticos donde se incluye, pero no se limita, a los principios de buena fe, transparencia y facilitación en donde se indica que las personas o entidades que recolecten o traten la información deben hacerlo siempre con buenas intenciones y permitiendo el acceso a esta información a quien lo requiera[32].

Como mecanismo de protección a la privacidad de la información y el acceso a sistemas informáticos, se realizó la modificación al Código Penal colombiano, mediante la Ley 1273 del 2009, la cual incluye las penas con las que se castigaría a las personas que violen la privacidad o seguridad de los datos contenidos en dichos sistemas[33].

2.3.2. Políticas de tratamiento de datos de la red social Twitter

Al registrarse en la red social Twitter, el usuario está aceptando la aplicabilidad de los terminos del servicio, la política de privacidad de datos y las reglas establecidas para dicha red social.

En estos documentos legales se reafirma al usuario que él es dueño de todo el contenido que pública y que al publicarlo a través de esta red social, le otorga a la misma una licencia de uso y distribución mundial sin retribuciones económicas para el autor. También se indica al usuario que no compartirá información que sea identificable a terceros salvo lo mencionado en la política de privacidad.

> "Al enviar, publicar o mostrar contenido a través de los servicios, nos otorga una licencia mundial, no exclusiva, libre del pago de derechos (con derecho a sublicencia) para usar, copiar, reproducir, procesar, adaptar. modificar, publicar, transmitir, mostrar y distribuir dicho contenido en todos y cada uno de los medios de comunicación o métodos de distribución posibles (conocidos ahora o desarrollados con posterioridad). Esta licencia nos autoriza a poner su contenido a disposición del resto del mundo y a permitir que otros hagan lo mismo. Usted acepta que esta licencia incluye el derecho de Twitter a proporcionar, promover y mejorar los servicios y a poner el contenido enviado a o a través de los servicios a disposición de otras empresas, organizaciones o personas para la sindicación, emisión, distribución, promoción o publicación de dicho contenido en otros medios y servicios, sujeto a nuestros términos y condiciones para el uso de dicho contenido. Dichos usos adicionales por parte de Twitter u otras empresas, organizaciones o personas pueden realizarse sin abonarle a usted una compensación con respecto al

[32] "LEY 1712 DE 2014". En: *Diario Oficial 49084 de marzo 6 de 2014* (2014). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=56882.

[33] "LEY 1273 DE 2009". En: *Diario Oficial 47.223 de enero 5 de 2009* (2009). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492.

contenido que haya enviado, publicado, transmitido o puesto a disposición pública de cualquier otra forma a través de los servicios."[34]

Por su parte en la política de privacidad el usuario acepta, para el caso de usuarios con residencia en Colombia darle potestad a la empresa *Twitter International Company* para almacenar, transferir y usar su información en cualquier parte del mundo donde la empresa mencionada o sus filiales operen. De igual manera se da a conocer al usuario que la información públicada se puede usar por servicios de terceros, y se recalca que Twitter no comparte datos los sensibles del usuario como los correspondientes al inicio se sesión.

"**Tuits, gente que sigue, listas, perfil y otra información pública:** Twitter está principalmente diseñado para ayudarle a compartir información con el mundo. La mayoría de la información que usted nos facilita a través de Twitter es información que nos está pidiendo que hagamos pública. Puede facilitarnos información de perfil para hacerla pública en Twitter, como por ejemplo, una breve biografía, su ubicación, su sitio web, fecha de nacimiento, o una fotografía. Además, su información pública incluye los mensajes que tuitea; los metadatos facilitados con los tuits, tales como cuándo ha tuiteado y la aplicación cliente que utilizó para tuitear; información sobre su cuenta, como el momento de su creación, el idioma, el país y la zona horaria; y las listas que crea, las personas a las que sigue, los tuits que retuitea o marca como "Me gusta", y las emisiones de Periscope en las que hace clic o con las que se relaciona de alguna forma (por ejemplo, haciendo comentarios o clic en el icono de corazón) en Twitter.

Twitter disemina amplia e instantáneamente su información pública a una amplia gama de usuarios, clientes y servicios, incluyendo motores de búsqueda, desarrolladores y editores que integran contenido de Twitter en sus servicios y organizaciones, tales como universidades, agencias de salud pública y empresas de investigación de mercado que analizan la información en busque de tendencias y conocimiento. Cuando comparta información o contenidos como fotografías, videos y enlaces a través de los servicios, debería reflexionar cuidadosamente sobre aquello que está publicando. Podemos utilizar esta información para sacar conclusiones, por ejemplo sobre los temas que le pueden interesar. Por defecto, casi siempre publicamos la información que usted nos facilita a través de los servicios de Twitter hasta el momento en que usted la elimine, pero generalmente ponemos a su disposición configuraciones o características, como tuits protegidos, para hacer que la información sea más privada si usted quiere. Para ciertos campos de información de perfil le ofrecemos ajustes de visibilidad para seleccionar quién puede ver esta información en su perfil. Si nos proporciona información de perfil y no puede ver un ajuste de visibilidad, esa información es pública. Puede modificar

[34]Twitter International Company. *Terminos del Servicio*. 2016. URL: `https://twitter.com/tos?lang=es` (visitado 11-06-2017).

el idioma y la zona horaria asociada con su cuenta en cualquier momento usando la configuración de su cuenta."[35]

[35]Twitter International Company. *Política de Privacidad de Twitter*. 2017. URL: `https://twitter.com/privacy?lang=es%7B%5C#%7Dupdate` (visitado 11-06-2017).

3. METODOLOGÍA

3.1. METODOLOGÍA DE DESARROLLO

Para el proyecto de investigación se analizó si se requeriría el uso de una metodología ágil, y así definir las consecuencias de su posible implementación; de este proceso se logró concluir que la mayor representación de costo es aportada por el tiempo que supone la realización del proyecto.

La definición de la metodología se inicia delimitando el alcance que va a tener la investigación, y descomponiendo la misma en problemas más pequeños, para así plantear una estructura desglosada que permita generar un plan de acción en el proyecto.

Según los resultados de dicho análisis se determinó que se deberá priorizar la gestión de riesgos en la realización del proyecto debido a la alta probabilidad de cambios en los requerimientos del mismo; de acuerdo a esto se utilizará una metogología ágil que se ajuste a los principios planteados por el manifiesto por el desarrollo ágil de software[1].

La metodología ***X.P.*** *Extreme Programming* es una metodología ágil que puede ser aplicada en los casos en que se requiera gestionar un alto riesgo de cambio en los requerimientos mediante la comunicación constante entre todos los participantes del proceso de desarrollo en donde se incluye al cliente como parte activa. Para comprender mejor el proceso involucrado en la metodología $X.P$ se dividió en varias etapas de la siguiente forma[2]:

1. Exploración.
2. Planificación de la entrega.
3. Iteraciones.
4. Producción.

[1] Kent Beck y col. *Manifiesto por el Desarrollo Ágil de Software*. URL: `http://agilemanifesto.org/iso/es/manifesto.html` (visitado 06-08-2017).

[2] Patricio Letelier y Mª Carmen Penadés. "Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". En: (). URL: `http://users.dsic.upv.es/asignaturas/eui/lds/doc/masyxp.pdf`.

5. Mantenimiento.

6. Muerte del proyecto.

Según Wells[3] para la implementación de *X.P.* hay que seguir una serie de reglas que se involucran en las etapas del proceso de desarrolo[4] así:

- **Planificación:** con base a las historias de usuario que son provistas por el cliente, se realiza una planeación de como se deberá realizar el proceso de desarrollo para la entrega que se va a empezar y se definen los recursos que se usarán en dicha iteración; en esta etapa se realizan las mediciones de avance del proyecto.

- **Diseño:** se realizan diseños sencillos basados en las historias de usuario y los posibles problemas que se podrían presentar en la codificación de la histora de usuario que se está atacando, así se disminuyen los riesgos de fallo en la estimación del tiempo que tomará desarrollarla.

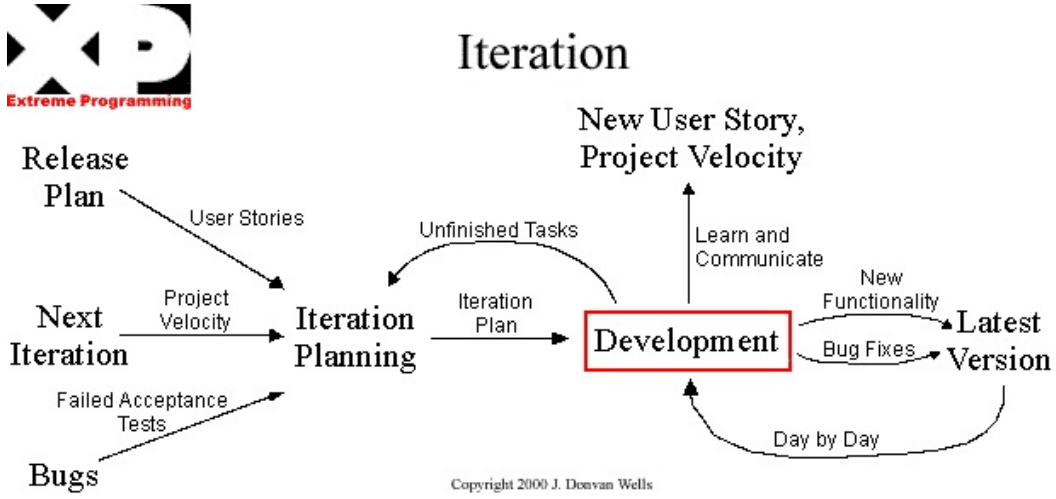
- **Codificación:** en esta etapa se deben programar primero pruebas unitarias para agilizar la implementación de los requerimientos dentro del código entregable, el cual a su vez deberá cumplir con los estándares de desarrollo. En esta etapa el cliente siempre estará disponible para complementar la información que podría ser requerida por los desarrolladores.

- **Pruebas:** cuando se encuentran errores en el software entregado se deberá crear una prueba de aceptación, en donde el cliente deberá describir detalladamente como se presentó la falla, y el equipo desarrollador deberá realizar pruebas unitarias antes de depurar para tener claridad de la parte del proceso en la cual se presenta la falla y así corregirla con mayor precisión.

[3] Don Wells. *Extreme Programming Rules*. URL: `http://www.extremeprogramming.org/rules.html` (visitado 09-08-2017).

[4] Álvaro Galván Lucas y Jose Manuel Torres Púa. *Extreme programming*. URL: `http://osl2.uca.es/wikiCE/index.php/Extreme%7B%5C_%7Dprogramming` (visitado 09-08-2017).

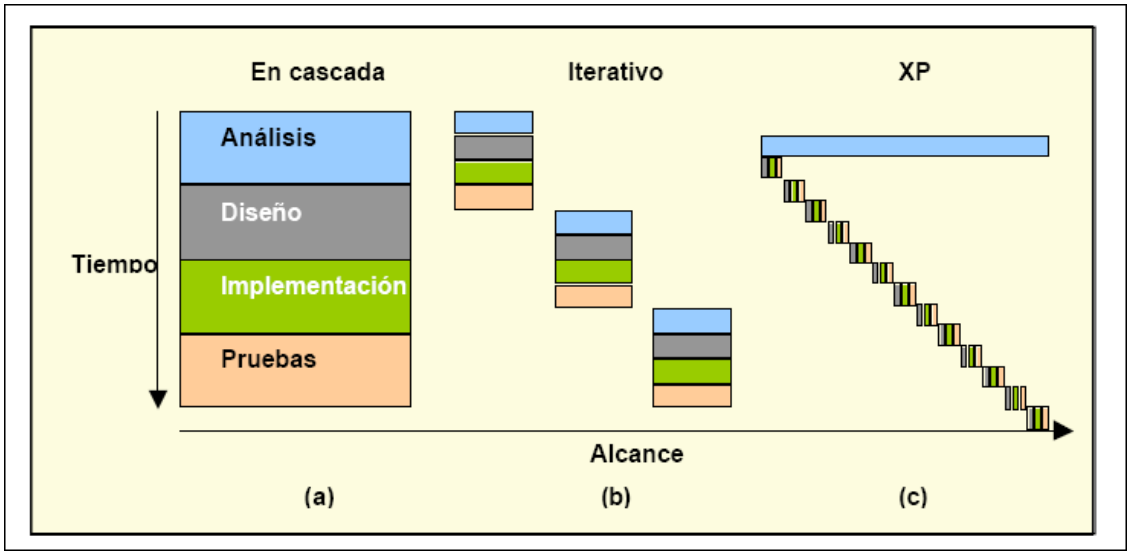
Figura 3-1.: Flujo del proceso de desarrollo en la metodología XP [48]



De acuerdo a estas fases, se debe que tener en cuenta que la metodología tiene diferentes iteraciones que se realizan de acuerdo a las revisiones y avances que la investigación vaya teniendo, además el proceso se debería complementar con un fuerte elemento de retroalimentación y análisis continuo.

Como resultado, se tendrá una etapa de análisis que será paralela a la realización del proyecto, y las etapas de desarrollo deben ser cortas e iterativas, para tener un resultado aproximadamente así:

Figura 3-2.: Descripción de las etapas de desarrollo en el tiempo usando XP [15]



3.2. CRONOGRAMA DE ACTIVIDADES

El cronograma de actividades se dividió en fases que concuerdan con la implementación de la metodología *Extreme Programming*, a estas fases se añadió la fase inicial llamada *"Planteamiento del proyecto"* en la que se incluyó el tiempo utilizado en el desarrollo del anteproyecto de grado.

Figura 3-3.: Diagrama de Gantt con el cronograma de desarrollo del proyecto.

Fase	Nombre	Duración	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16	Semana 17	Semana 18	Semana 19	Semana 20	Semana 21	Semana 22	Semana 23	Semana 24	Semana 25	Semana 26
Planteamiento del proyecto	Formulación del anteproyecto	2 meses																										
	Correcciones al documento de anteproyecto	1 mes																										
Exploración	Definición inicial de historias de usuario de la herramienta	1,5 sem																										
	Definición de restricciones tecnológicas del proyecto	1,5 sem																										
	Planteamiento de la arquitectura de la aplicación	1,5 sem																										
	Planteamiento de los métodos de integración de la herramienta	1,5 sem																										
Planificación de la entrega	Establecer las prioridades de las historias de usuario para la entrega inicial	1 sem																										
Iteraciones	Planificación	2,5 meses																										
	Diseño	2,5 sem.																										
	Codificación	2,5 sem.																										
	Pruebas	2,5 sem.																										
	Diseño	2,5 sem.																										
	Codificación	2,5 sem.																										
	Pruebas	2,5 sem.																										
Producción	Pruebas entrega inicial del proyecto	1 sem																										
	Diseño entrega inicial del proyecto	1 sem																										
	Codificación entrega inicial del proyecto	1 sem																										
Mantenimiento	Diseño	2 sem																										
	Codificación	2 sem																										
	Pruebas	2 sem																										
Muerte	Cierre del proyecto	2 sem																										

3.3. PRODUCTOS A ENTREGAR

Durante la realización del proyecto se generarán entregables de diferentes tipos como lo son el sistema de software finalizado y los documentos relacionados en su diseño, a continuación se presenta el listado de entregables:

Tabla 3-1.: Listado de entregables del proyecto

PRODUCTOS A ENTREGAR		
TIPO	**NOMBRE**	**FECHA DE ENTREGA**
Software	Beta Classification	17 de mayo de 2018
Documento	Caracterización del estado del arte	01 de abril de 2018
Documento	Documento de arquitectura de software	01 de mayo de 2018
Documento	Manual de usuario	17 de mayo de 2018
Documento	Manual de administrador	17 de mayo de 2018

3.4. PRESUPUESTO DEL TRABAJO

Se generó la especificación para los costos totales del proyecto en rubros de interés los cuales se presentan en la siguiente tabla:

Tabla 3-2.: Costos de realización del proyecto

PRESUPUESTO GLOBAL DEL ANTEPROYECTO Y PROYECTO DE GRADO				
COSTOS DIRECTOS				
Descripción	**Medida**	**Costo unitario**	**Cantidad**	**Total**
Recurso humano: estudiante de ingeniería de sistemas	Hora	$11.000	384	$4.224.000
Equipo: computador portátil	Unidad	$1.000.000	1	$1.000.000
Taxis y buses	Recorrido	$3.650	96	$350.400
Materiales: suscripciones a bases de datos	Unidad	$200.000	1	$200.000
Materiales: papelería	Unidad	$500.000	1	$500.000
COSTOS INDIRECTOS				
Descripción	Medida	Costo unitario	Cantidad	Total
Servicios públicos: Energía	Kw	$600	185	$111.000
Servicios públicos: Internet domiciliario	Horas	$100	384	$38.400
Imprevistos	Porcentaje	$64.238	20	$1.277.080
COSTO TOTAL DEL PROYECTO				**$7.700.880**

3.5. ESTRATEGIAS DE COMUNICACIÓN Y DIVULGACIÓN

Con el fin de dar visibilidad a los resultados obtenidos mediante este proyecto investigativo, se busca hacer uso de los espacios para socialización de trabajos de grado y resultados de los grupos de investigación que genera la Universidad Católica de Colombia; además la universidad cuenta con el repositorio institucional *"Repositorio Institucional Universidad Católica de Colombia"* en donde se almacenan los trabajos de grado aprobados.

Por último, el uso de las redes sociales brinda visibilidad entre los contactos de quienes compartan estos resultados, por lo tanto también se podría tener en cuenta como un medio para la comunicación y divulgación.

4. ESTADO DEL ARTE

Para la descripción del estado del arte se revisó literatura publicada entre los años 2014 y 2018, y se ha encontrado, que las técnicas de clasificación de textos cortos son ampliamente utilizadas para el análisis de sentimientos[1], clasificación de titulares de prensa[2], entre otros[3]. En las tareas de clasificación de textos las categorías en las cuales se clasificarán los datos de entrada varían, según los datos provistos en el dataset de entrenamiento; para la realización de este proyecto, se desea validar la pertenencia a ciertas tendencias orientadas a la paz o a la guerra.

Según Song[4], la clasificación de textos cortos representa un reto debido a la posible pérdida de precisión que supone el no contar con la cantidad necesaria de información, esto se suma a la aparición de ruido y términos del argot dentro de las variables que se están estudiando.

Esta situación conlleva a que el algoritmo trabaje con datos que no se encuentran estandarizados, por lo que los métodos de clasificación de textos tradicionales tienden a fallar ya que suelen hacer uso de las similaridades en la frecuencia de términos, ignorando las características de los textos cortos.

Partiendo de esto, el autor indica que el proceso de clasificación de textos cortos deberá

[1] Xuetong Chen y col. "What About Mood Swings: Identifying Depression on Twitter with Temporal Measures of Emotions". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1653-1660. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191624. URL: https://doi.org/10.1145/3184558.3191624.

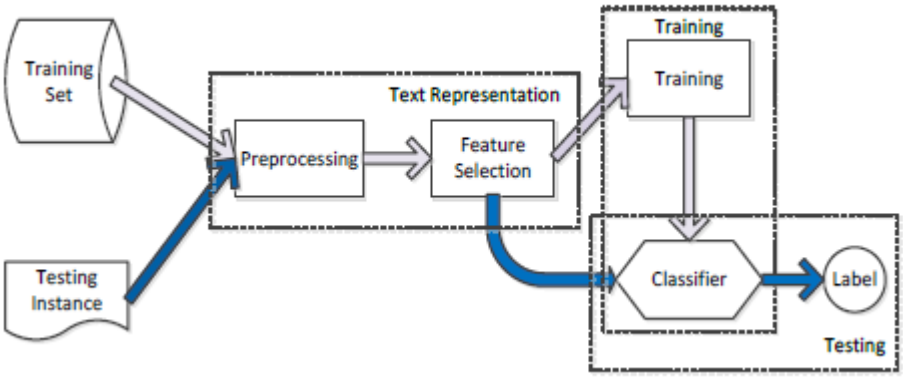
[2] Yuanxin Ouyang y col. "News Title Classification with Support from Auxiliary Long Texts". En: *Neural Information Processing*. Ed. por Chu Kiong Loo y col. Cham: Springer International Publishing, 2014, págs. 581-588. ISBN: 978-3-319-12640-1.

[3] Jian Xu y col. "Signature based trouble ticket classification". En: *Future Generation Computer Systems* 78 (2018), págs. 41-58. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.07.054. URL: http://www.sciencedirect.com/science/article/pii/S0167739X16308056, Walaa Medhat, Ahmed Hassan y Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". En: *Ain Shams Engineering Journal* 5.4 (2014), págs. 1093-1113. ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2014.04.011. URL: http://www.sciencedirect.com/science/article/pii/S2090447914000550.

[4] Ge Song y col. "Short Text Classification: A Survey". En: *Journal of Multimedia* 9.5 (2014), págs. 635-643. ISSN: 1796-2048. DOI: 10.4304/jmm.9.5.635-643. URL: http://ojs.academypublisher.com/index.php/jmm/article/view/12635.

contar con una etapa de extracción de características previa a las etapas de entrenamiento o clasificación como se ve en la figura 4-1.

Figura 4-1.: Flujo del proceso de clasificación de textos cortos [44]



Otros autores como Singh y Kumari, reconocen la importancia del pre procesamiento de los datos en la calidad de los resultados obtenidos a partir de estos datasets ya que al ser sometidos a una etapa de normalización previa a la puesta en marcha de los algoritmos, se logran obtener distinciones más significativas entre las clases, lo cual conlleva de forma inherente a una mejora en los resultados de las tareas de clasificación[5].

McCartney[6] realiza la comparación del impacto que tienen la longitud de los textos a estudiar y las etapas de pre procesamiento, extracción de características durante el proceso de clasificación, para tal fin genera diez subconjuntos de la información a clasificar según el tipo de pre procesamiento que le aplica a los mismos, dentro de los cuales se encuentran.

- Texto original: No se le realiza ninguna mejora al dato que se está evaluando.
- Texto limpio: Se realiza la remoción de signos de puntuación como puntos aparte, urls y otros componentes que puedan generar ruido al momento de realizar la clasificación.
- Texto lematizado: Se aplica un proceso de lematización al texto que se quiere clasificar para dejar únicamente sus raíces.

[5]Tajinder Singh y Madhu Kumari. "Role of Text Pre-processing in Twitter Sentiment Analysis". En: *Procedia Computer Science* 89 (2016). Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India, págs. 549-554. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.06.095. URL: http://www.sciencedirect.com/science/article/pii/S1877050916311607.

[6]Austin Mccartney. *"How Short is a Piece of String?": An Investigation into the Impact of Text Length on Short-Text Classification Accuracy*. 2017.

- Bigramas: Sobre el texto lematizado se seleccionan los bigramas presentes.

Posterior a la etapa de pre procesamiento se aplicaron tres algoritmos de clasificación sobre los diez datasets derivados los cuales son:

- Clasificadores ingenuos de Bayes.
- Máquinas de soporte Vectorial.
- Análisis semántico latente.

Luego de la aplicación de las tareas de clasificación se encontró que en todos los casos la longitud de los textos a usar tiene incidencia en los resultados obtenidos, sin embargo, los procesos que se realizan en la etapa del pre procesamiento de datos son fundamentales para la mejora de los resultados. Además se evidenció que el mejoramiento a través de la búsqueda de bigramas conlleva a un aumento significativo en la precisión de los resultados obtenidos, en las figuras 4-2, 4-3 y 4-4 el autor ilustra los resultados obtenidos en función del tratamiento realizado sobre los textos y la longitud de los mismos.

Figura 4-2.: Precisión de los resultados obtenidos a través del clasificador ingenuo de Bayes con respecto a la longitud de los datos de entrada y el pre procesamiento realizado [31]

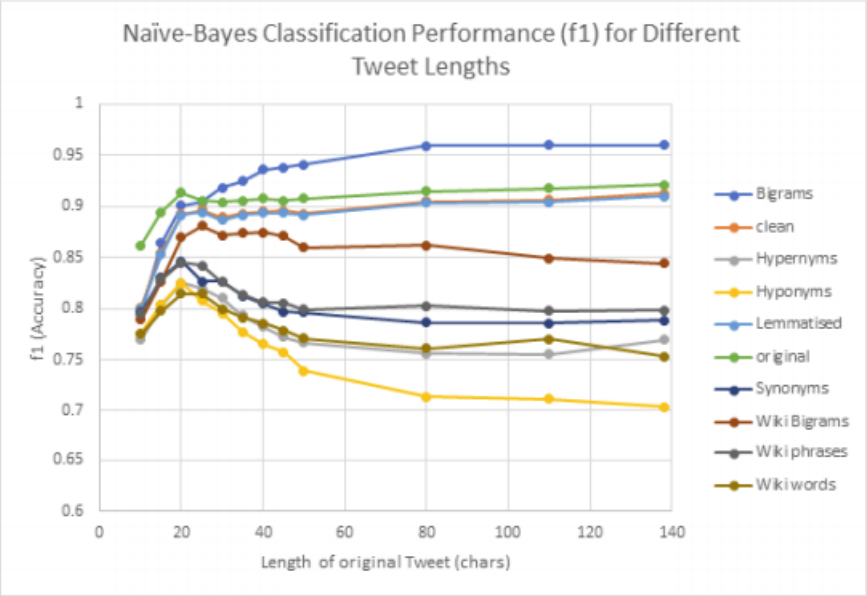


Figura 4-3.: Precisión de los resultados obtenidos a través del clasificador máquina de soporte vectorial con respecto a la longitud de los datos de entrada y el pre procesamiento realizado [31]

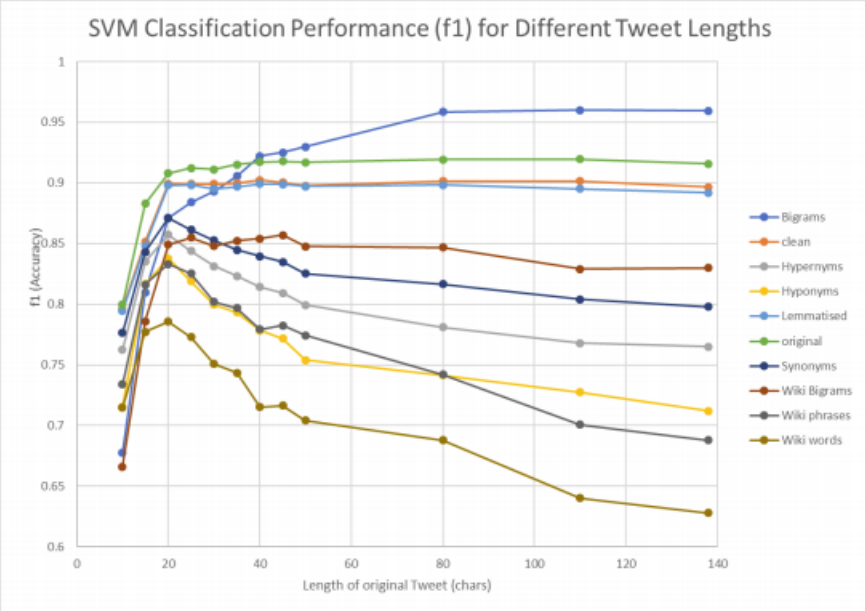
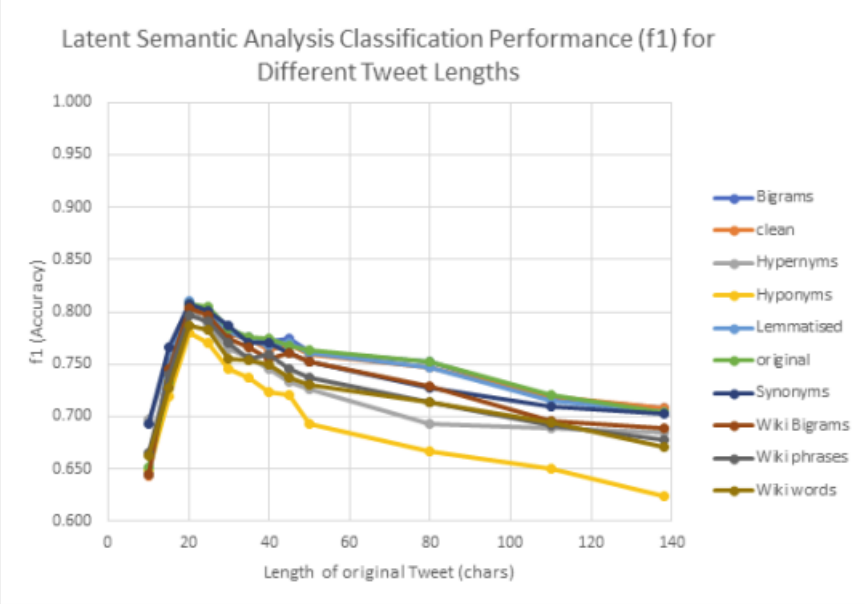


Figura 4-4.: Precisión de los resultados mediante análisis semántico latente con respecto a la longitud de los datos de entrada y el pre procesamiento realizado [31]



Por su parte Kumar[7] realiza las tareas de clasificación a través de la comparación de similitudes usando procesamiento de texto *(SMTP por sus siglas en inglés)*, para este tipo de clasificación tradicionalmente se tiene en cuenta únicamente cuando la similitud es positiva (presencia de similitudes) o negativa (ausencia de similitudes), sin embargo, el autor plantea un estado en el cual se presenta una posición neutra del dato de entrada, es decir, cuando las características que se obtuvieron del dato no son determinantes al momento de realizar la comparación con lo aprendido en la etapa de entrenamiento, esto se realiza con el ánimo de aumentar la precisión que entregan los algoritmos de clasificación, en este caso los K-vecinos más cercanos del inglés *K-Nearest Neighbors (KNN)*, en donde K representa una cantidad de clases vecinas contras las cuales se comparan los datos que se estén estudiando.

En la etapa de pre procesamiento, se realiza limpieza de los mismos, removiendo las URL, signos de puntuación y variables que puedan generar ruido al proceso, luego se realiza un proceso de lematización y finalmente se procede con la etapa de aprendizaje, y verificación del comportamiento de los algoritmos, acá se varía el valor de K entre 3 y 15 clases vecinas, y se comparan los resultados de la clasificación con los clasificadores de máquinas de soporte vectorial, árboles aleatorios y el mismo KNN tradicional. En cada caso, se comprobó que las modificaciones propuestas por el autor incrementan la precisión obtenida en las tareas de clasificación.

Ali[8] propone también un framework probabilístico similar al de McCartney, el cual se compone de tres etapas:

- Pre procesamiento: El texto de entrada pasa por un proceso de desensibilización y limpieza en donde se remueven todos los datos personales que se puedan encontrar y las variables que podrían generar ruido, teniendo en cuenta que como caso de estudio se trabaja sobre titulares de noticias de diferentes categorías.

  Luego de esto, se seleccionan todos los bigramas y trigramas posibles dentro de los datos.

- Entrenamiento: Se generan matrices de probabilidad de pertenencia a clases según la ocurrencia de las palabras sin agrupar, bigramas y trigramas para cada clase estudiada, y basados en este modelo de procede a la última etapa.

[7] H M Keerthi Kumar y col. "Classification of Sentiments in Short-text: An Approach Using mSMTP Measure". En: *Proceedings of the 2Nd International Conference on Machine Learning and Soft Computing*. ICMLSC '18. Phu Quoc Island, Viet Nam: ACM, 2018, págs. 145-150. ISBN: 978-1-4503-6336-5. DOI: 10.1145/3184066.3184074. URL: http://doi.acm.org/10.1145/3184066.3184074.

[8] Mubashir Ali y col. "A probabilistic framework for short text classification". En: *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. Ene. de 2018, págs. 742-747. DOI: 10.1109/CCWC.2018.8301712.

- Clasificación usando modelos probabilísticos: Con las matrices generadas en la etapa anterior se estudian los datos de entrada, asignando como categoría resultante la categoría con mayor probabilidad de acierto.

En la etapa experimental el autor realiza la verificación de la precisión obtenida para cada una de las tres estrategias de agrupación aplicadas en la etapa de pre procesamiento, reafirmando los resultados obtenidos por McCartney ya que la estrategia que dio mejores resultados fue la del descubrimiento de bigramas con un 92,41 % de precisión, seguido por las palabras que no se encuentran agrupadas con 84,25 % y para terminar vemos como la agrupación por trigamas ocasiona una pérdida de precisión significativa para este estudio con un 79,22 % obtenido. Todos los modelos fueron entrenados con 8000 datos por cada clase, y con 2000 datos de comprobación, el porcentaje de precisión se calcula con el número de datos clasificados correctamente dividido entre la totalidad de los datos de comprobación.

Ghosh y Desarkar[9] proponen que las tareas de clasificación se hagan basados en la existencia de ciertos términos clave, sin prestar mayor atención a las clases provistas durante la etapa de entrenamiento de los algoritmos, ya que buscan determinar la pertenencia a clases dependiendo de la existencia de términos excluyentes, la etapa de pre procesamiento se realiza también en su propuesta a través de lematización de los datos de entrada y limpieza de los mismos, sin embargo los resultados obtenidos demuestran que en este modelo se depende en gran medida de la calidad y la cantidad de datos que se tengan disponibles de cada una de las clases que se quieran estudiar, ya que de no contar con datasets de entrenamiento que cumplan con estas condiciones se puede llegar a presentar una pérdida en la precisión de las clasificaciones realizadas, llegando a reportar en el peor de los casos 65 % de precisión en las predicciones realizadas.

En el trabajo realizado por Wang[10] se propone la creación de un framework ligero para la clasificación de textos cortos que puede ser usado en sistemas de recomendación en línea como buscadores, en este framework se propone reemplazar el uso del modelo de bolsa de palabras, el cual se centra en la clasificación de los textos según la similitud que presenten en la ocurrencia de determinadas palabras conocidas previamente, a las cuales se les ha asignado un puntaje que servirá para medir la similitud entre textos de la misma categoría[11] por el

[9] Samujjwal Ghosh y Maunendra Sankar Desarkar. "Class Specific TF-IDF Boosting for Short-text Classification: Application to Short-texts Generated During Disasters". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1629-1637. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191621. URL: https://dl.acm.org/citation.cfm?id=3191621.

[10] Fang Wang y col. "Concept-based Short Text Classification and Ranking". En: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14* (2014), págs. 1069-1078. DOI: 10.1145/2661829.2662067. URL: http://dl.acm.org/citation.cfm?doid=2661829.2662067.

[11] Jason Brownlee. *A Gentle Introduction to the Bag-of-Words Model*. 2017. URL: https : / /

modelo de bolsa de conceptos en el cual se busca relacionar palabras con categorías contextuales más amplias, se da un ejemplo para las palabras *"Honda"* y *"Jeep"* las cuales si bien no están relacionadas a través de su significado, se encuentran estrechamente relacionadas por su contexto, ya que ambas pueden ser enmarcadas dentro la categoría autos.

Sin embargo, este modelo requiere de la creación de los conceptos para poder ser puesto en funcionamiento, este proceso se realiza mediante el aprovechamiento de bases de datos de conceptos pre existentes, una vez se obtienen los conceptos aprovechables, se procede con la agrupación de conceptos según su relación, para ejemplificar este paso el autor propone la agrupación de los conceptos de nación, país desarrollado, país en vías de desarrollo en la categoría país, buscando poder determinar sub categorías en la predicción de una categoría "padre", o relacionar conceptos que no tienen relación aparente.

El proceso realizado para la clasificación de textos haciendo uso de este framework tendría las siguientes etapas:

- Aprendizaje del modelo de conceptos: En esta etapa se busca que el algoritmo aprenda del dataset de entrenamiento las relaciones que tienen las determinadas clases de entrada con los conceptos definidos, y como afecta la aparición de los mismos en la pertenencia del dato a la clase, el autor divide esta etapa en las siguientes tareas.
  - Reconocimiento de entidades.
  - Generación de candidatos.
  - Puntuación de conceptos.
- Conceptualización del texto a clasificar: En esta etapa se busca remover la ambigüedad que se pueda presentar en el mismo, mediante la contextualización de los conceptos, ya que en el ejemplo propuesto por el auto en el texto corto "apple ipad"la palabra "apple" puede referirse a una compañía o a una fruta, por lo que la palabra "ipad" es usada para contextualizar "apple" al significado de la compañía, dotando así de sentido al texto, y removiendo el problema de ambigüedad de algunos de los conceptos presentes en el mismo.
- Clasificación y posicionamiento: La última etapa propone la asignación de una clase para el dato de entrada que se esté clasificando mediante la aproximación de similitudes con los conceptos de clase, acá es donde se evidencian los umbrales de permanencia a las clases definidos mediante la asignación de pesos a determinados conceptos, ya que para obtener resultados con mayor precisión el autor recomienda refinar este proceso de asignación de peso.

`machinelearningmastery.com/gentle-introduction-bag-words-model/` (visitado 21-04-2018).

Después de que se determina la similitud que tiene el texto de entrada con las clases definidas se dice que el mismo pertenece a la clase con la que tenga mayor afinidad, y se brindan demás clases a las que "podría" pertenecer el dato de entrada con diferentes probabilidades de pertenencia, este proceso se conoce como la fase de posicionamiento, dentro de la cual se destacan dos posibles usos comunes:

- Posicionamiento por similitud: Este uso es el más común, ya que busca entregar un listado de clases con las que el dato de entrada tiene cierta similitud, por lo general se entrega un listado de las clases ordenadas de manera descendente según el posicionamiento asignado por el algoritmo recorriendo desde la clase con mayor grado de similitud a la menor.

- Posicionamiento con diversidad: Este tipo de clasificación es propuesto por los autores como óptima para sistemas de recomendaciones en linea, como buscadores de noticias ya que se busca entregar un listado de recomendaciones que si bien no tienen relación directa con los datos de entrada, deben estar relacionados por algún concepto padre, este proceso conlleva en las tareas de clasificación de textos pasos adicionales para poder obtener diferentes subtemas que se relacionen con el dato de entrada, sin embargo como se menciona en la descripción del framework, los algoritmos que hagan uso de este requieren hacer uso de bases de datos de conceptos, mediante este proceso de uso se extraen agrupaciones o relaciones entre temas, lo cual favorece la ejecución en tiempo real de dichas tareas de recomendación.

En la etapa experimental se determinó que la precisión de los resultados está fuertemente ligada con la calidad y la cantidad de la información disponible en la base de datos de conceptos que se use de referencia, ya que desde esta es desde donde se generan los modelos que permiten asignar los datos a las categorías, por lo tanto, partiendo de una base de datos consistente, se obtuvieron resultados con un promedio de precisión del 90 % sobre las tareas de clasificación de textos cortos.

Di Nunzio, Maistro y Vezzani[12] hacen un especial énfasis en los costos asociados que conlleva la clasificación manual por parte de expertos de los datos de entrenamiento, y ya que este proceso es inherente a las tareas de clasificación en general que se deseen realizar a través de algoritmos y/o técnicas de aprendizaje supervisado, proponen la apertura de la etapa de entrenamiento a la comunidad en general a través de técnicas de gamificación que permitan que personas que no son expertas interactúen con los algoritmos de clasificación, mediante

[12]Giorgio Maria Di Nunzio, Maria Maistro y Federica Vezzani. "A Gamified Approach to NaiVe Bayes Classification: A Case Study for Newswires and Systematic Medical Reviews". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1139-1146. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191547. URL: https://doi.org/10.1145/3184558.3191547.

la modificación de parámetros propios del algoritmo.

En este caso los experimentos fueron realizados sobre clasificadores ingenuos de Bayes, en donde se permitió que "jugadores" trataran de ubicar una línea separadora entre las diferentes clases que se desean clasificar, de esta forma se reescriben con cada modificación de dicha línea los parámetros tenidos en cuenta por el clasificador para determinar la pertenencia a una de las dos clases estudiadas.

Según los resultados expuestos por los autores se evidencia que en casos donde la tarea de clasificación presenta un nivel de dificultad mayor, la precisión de los algoritmos entrenados por los "jugadores" se comporta de manera similar, y en algunas ocasiones mejor que la obtenida por los clasificadores ingenuos de Bayes y las máquinas de soporte vectorial. Sin embargo en el caso promedio, si bien los resultados obtenidos son buenos y cercanos a los obtenidos por los demás clasificadores, siguen siendo inferiores a los obtenidos de manera tradicional.

Dentro de las etapas del experimento realizado por los autores, se recompensó a los participantes que lograran obtener mejor desempeño con la menor cantidad de recursos, obteniendo de esta forma un uso aproximado al 30 % de los datos de entrenamiento y manteniendo los niveles de precisión, en el enlace: `https://gmdn.shinyapps.io/Classification/`[13] se puede encontrar el experimento realizado, la figura 4-5 muestra una captura de pantalla de la interfaz gráfica propuesta dentro del experimento.

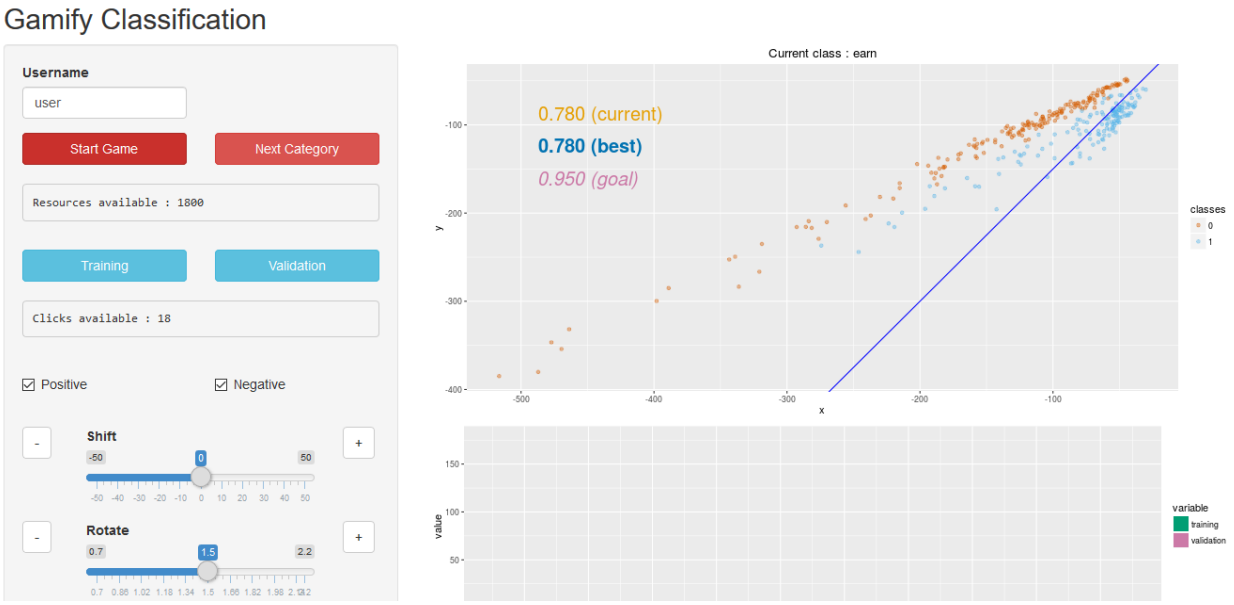
Dentro de los usos que se le dan a las tareas de clasificación de textos cortos Desmet y Hoste[14] proponen el seguimiento a través de publicaciones expuestas en redes sociales para detectar pensamientos suicidas, ya que este comportamiento se ha convertido en un problema de salud pública, sin embargo debido a la gran cantidad de datos expuestos se hace prácticamente imposible el seguimiento manual de los mismos. Teniendo en cuenta además las características propias de los textos cortos, se hace inviable limitar los datos mediante búsquedas de palabras clave sobre los contenidos producidos por los usuarios de las redes sociales. En la etapa del pre procesamiento se acude a la extracción de características buscando optimizar la precisión de los modelos de clasificación.

Los autores trabajaron en conjunto con un centro especializado para la prevención del suicidio en Bélgica para obtener una base de datos alimentada con casos reales y que se encontrara

[13] visitado 21-04-2018

[14] Bart Desmet y Véronique Hoste. "Online suicide prevention through optimised text classification". En: *Information Sciences* 439-440 (2018), págs. 61-78. ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2018.02.014`. URL: `http://www.sciencedirect.com/science/article/pii/S002002551830094X`.

Figura 4-5.: Interfaz gráfica del experimento realizado por los autores Di Nunzio, Maistro y Vezzani [13]



bien documentada, ya que al estar trabajando con modelos de aprendizaje supervisado requieren de un conjunto de datos de entrenamiento, a partir de los cuales se deben poder generar los correspondientes modelos de clasificación.

Sin embargo dado que los casos en los cuales se publican pensamientos o planes suicidas son reducidos, el problema debe ser abordado desde dos tareas de clasificación diferentes pero complementarias las cuales son:

- Detección de los pensamientos o planes suicidas: Se realizan tareas de clasificación de texto para determinar si el dato de entrada contiene o no elementos de alerta.
- Clasificación de la severidad del caso: En esta etapa del problema se analiza que tan grave es el caso según una escala definida por los investigadores que va desde baja, hasta alta, dependiendo de los elementos de alerta que exprese el dato de entrada.

Para la realización de este proyecto los investigadores hicieron uso del clasificador máquinas de soporte vectorial, con una etapa de pre procesamiento en la cual se incluye la lematización y limpieza de los textos que se van a estudiar; obteniendo como resultado un 92.08 % de precisión con respecto a las tareas de detección y un 83.75 % en las tareas de detección de criticidad de los hallazgos, estos valores corresponden al mejor de los casos en ambas tareas.

5. DESARROLLO

5.1. HISTORIAS DE USUARIO

Durante la etapa de levantamiento de requerimientos se definieron una serie de historias de usuario requeridas para la realización del proyecto, las cuales agrupan las funcionalidades, facilitando así su entendimiento por parte del equipo de desarrollo, las cuales se especifican en la tabla 5-1:

Tabla 5-1.: Listado de historias de usuario definidas para el componente

ID	**Historia de Usuario**
HU1	**COMO:** usuario **QUIERO:** Entrenar algoritmos a través de un dataset **PARA:** Poder hacer uso de los algoritmos entrenados en tareas de clasificación
HU2	**COMO:** usuario **QUIERO:** Cargar un dataset **PARA:** Poder entrenar los algoritmos que seleccioné
HU3	**COMO:** usuario **QUIERO:** Cargar modelos de algoritmos previamente entrenados **PARA:** Poder hacer uso de dichos modelos entrenados en tareas de clasificación
HU4	**COMO:** usuario **QUIERO:** Conocer el rendimiento de los algoritmos entrenados **PARA:** Que al entrenarlos a través de un dataset conozca el porcentaje de precisión obtenido por cada uno de ellos
HU5	**COMO:** usuario **QUIERO:** Cargar un dataset **PARA:** Que los algoritmos previamente entrenados, clasifiquen su contenido
HU6	**COMO:** usuario **QUIERO:** Conocer el tiempo invertido en la clasificación de un dataset por cada algoritmo **PARA:** Comparar los tiempos de respuesta de cada algoritmo
HU7	**COMO:** usuario **QUIERO:** Guardar los resultados de la etapa de entrenamiento de un algoritmo **PARA:** Poder cargalo nuevamente más adelante y omitir la etapa de entrenamiento

5.2. REQUERIMIENTOS

Para el desarrollo de este componente de software se definió una serie de requerimientos funcionales y no funcionales que deberá cumplir, los cuales se enuncian en la tabla 5-2 y 5-3 respectivamente.

Tabla 5-2.: Listado de requerimientos funcionales del componente.

Identificador	**Nombre**
RF_001	Cargar conjunto de datos.
RF_002	Definir atributo de salida.
RF_003	Entrenar clasificador Naive Bayes (NB).
RF_004	Entrenar clasificador Árbol de Decisión (DT).
RF_005	Entrenar clasificador Máquina de Soporte Vectorial (SVM).
RF_006	Entrenar clasificador Red Neuronal Artificial (ANN).
RF_007	Entrenar clasificador k-nearest neighbors (KNN).
RF_008	Generar gráficas de rendimiento.

Para los requerimientos de entrenamiento se deberá permitir seleccionar el nombre de la columna que contiene las clases y de la columna que contiene los datos (texto a clasificar).

Como salida de este proceso, se debe entregar un resumen en donde se identifique la precisión sumado a una gráfica que corresponda con la matriz de confusión generada para ese algoritmo en particular y el modelo entrenado, este modelo se deberá poder cargar dentro del componente para omitir la clasificación mediante los conjuntos de datos.

Por su parte la opción de clasificación debe permitiringresar el nombre de la columna que contenga los datos. Y como salida al proceso de clasificación se deberá entregar una gráfica en donde se pueda comparar el tiempo invertido por los clasificadores en el mismo dataset, y los dataset resultantes del proceso, con la clase predecida.

Tabla 5-3.: Listado de requerimientos no funcionales del componente.

Identificador	**Nombre**	**Atributo de calidad**
RNF_001	Tiempo de respuesta.	Rendimiento.
RNF_002	Facilidad de uso.	Usabilidad.
RNF_003	Documentación del componente.	Usabilidad.
RNF_004	Interacción con otros sistemas.	Extensibilidad.

El componente al ofrecer una interfaz gráfica debe ser amigable y de fácil uso, ya que se espera que sea usado por el público en general, quien no necesariamente debe tener conoci-

mientos avanzados para hacer uso del componente.

Para conocer el detalle de los requerimientos listados acá remítase al documento anexo SRS-1804-Componente de clasificación.

5.3. ARQUITECTURA

En cuanto a la arquitectura del componente se usó la arquitectura cliente servidor, y la comunicación se realiza a través de peticiones HTTP, en el documento anexo SAD-1804-Componente de clasificación detallan los componentes de dicha arquitectura, y además se detallan los puntos de vista:

- Punto de vista de Stakeholder: busca explicar los actores que intervinieron en la construcción del componente de clasificación.
- Punto de vista de función negocio: explica las funciones de los actores dentro de los procesos del componente de software.
- Punto vista de cooperación de la aplicación: contextualiza la funcionalidad de la aplicación y su distribución en el modelo lógico de la misma.
- Punto vista de uso de la aplicación: basados en un proceso central, que en este caso es la clasificación de textos cortos, se explican las funcionalidades que prestan cada uno de los módulos del componente.
- Punto vista de infraestructura: describe como se distribuyó el componente en cuanto a la infraestructura disponible.
- Punto vista de Organización e implementación: explica el rol que juega la infraestructura dentro de la aplicación, indicando las partes del componente que se soportan en ella.
- Punto vista de proyecto: describe la forma en como el proyecto se relaciona con los diferentes actores esenciales para su desarrollo.

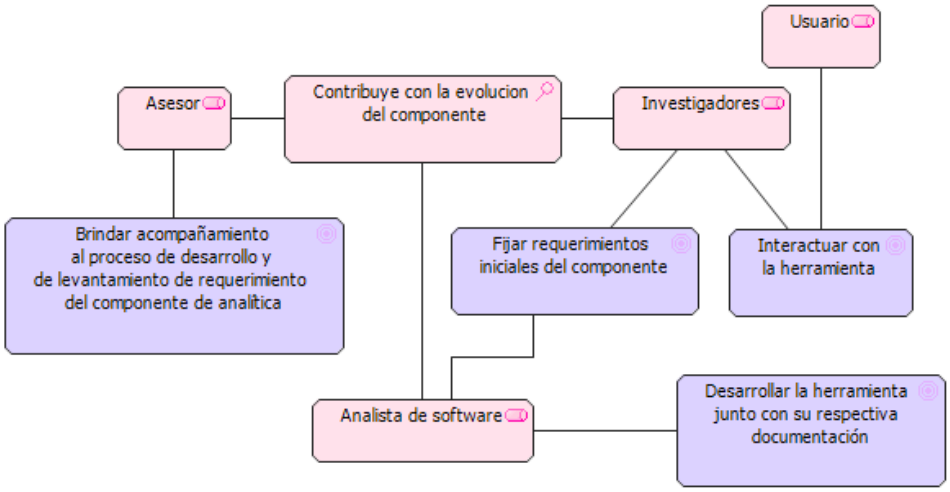
Para este proyecto los stakeholders son:

- Analista de software
- Asesor del trabajo de grado.
- Investigadores.

- Usuarios de la plataforma.

sus funciones son definidas en la figura 5-1.

Figura 5-1.: Stakeholders y roles dentro del desarrollo



Por su parte decimos que la funcionalidad central para el componente de software es la clasificación de textos cortos, y de esta se espera que sus módulos aporten al cumplimiento de esta tarea según lo ilustrado en la figura 5-2:

Figura 5-2.: Diagrama de uso de aplicación

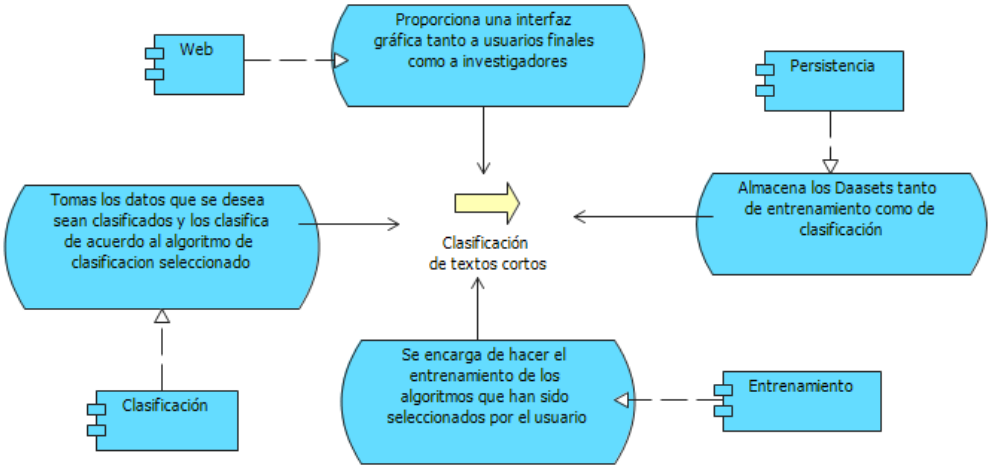


Figura 5-3.: Flujo del proceso para las operaciones de entrenamiento de los algoritmos

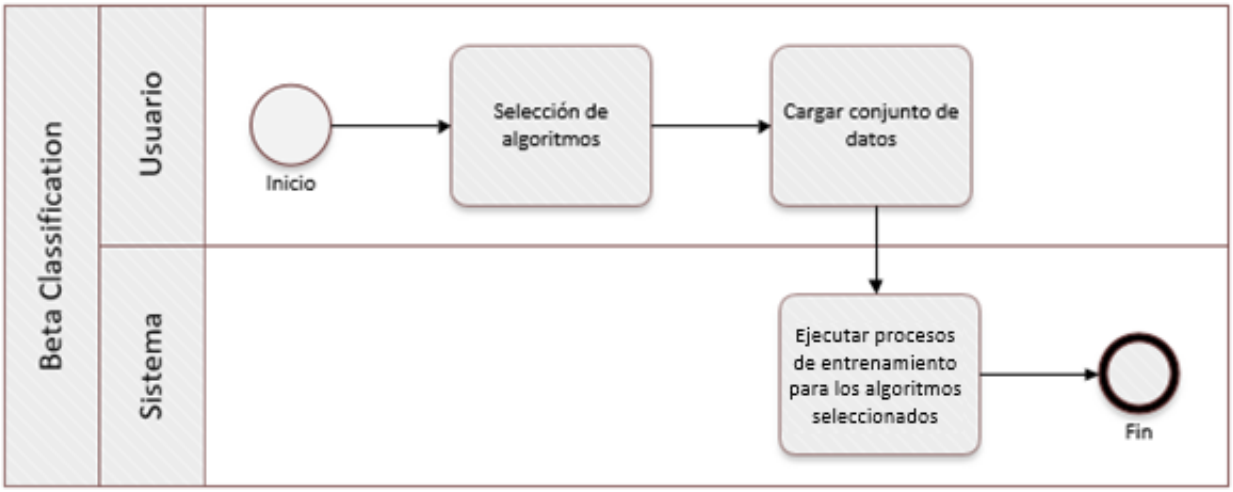
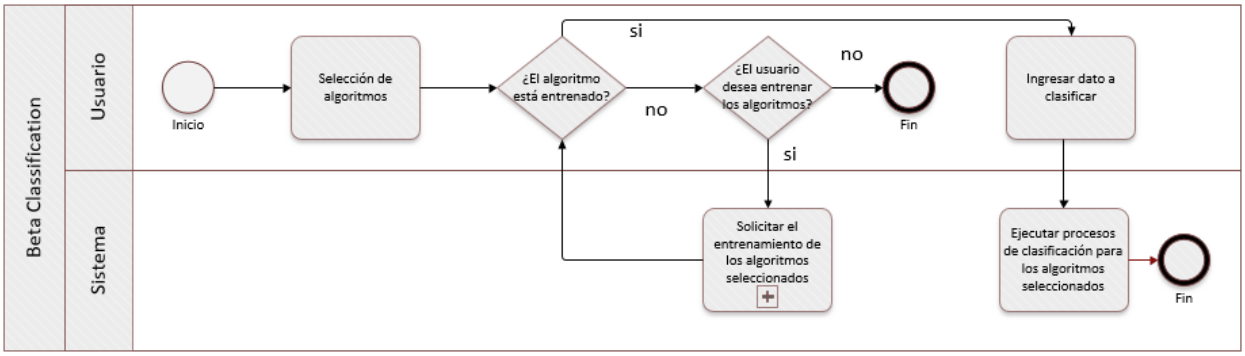


Figura 5-4.: Flujo del proceso para las operaciones de clasificación de los algoritmos



5.4. DISEÑO

Por su parte el documento anexo SDD-1804-Componente de clasificación se da una descripción de los flujos de proceso que puede tener la aplicación, y se detalla la interacción que se puede dar entre los usuarios y el componente, en este documento se especifica también como se harán las peticiones, teniendo en cuenta que el componente de software debe ser fácilmente integrable con plataformas web, en este caso en particular, se busca la integrabilidad con la plataforma propuesta como solución al proyecto de investigación "Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política".

Como se puede evidenciar en las figuras 5-3 y 5-4, en las cuales se exponen los diagramas de flujo para las operaciones de entrenamiento y clasificación, que forman parte indispensable para la funcionalidad del componente.

5.5. INSTALACIONES Y EQUIPO REQUERIDO

5.5.1. Instalaciones

Para la realización del proyecto se requerirá de las siguientes instalaciones:

- Instalaciones de la Universidad Católica de Colombia, incluyendo salones y salas de cómputo.
- Residencia personal con acceso a internet y servicios públicos.

5.5.2. Equipo requerido

En esta sección se busca describir los equipos requeridos directamente en la realización del proyecto.

- Computador usado para el desarrollo del proyecto incluyendo tareas de documentación y codificación:
  - Procesador Intel® Core™ i5 de cuarta generación.
  - Memoria RAM de 8 GB.
  - Al menos 200 GB de espacio libre en disco duro.
- Servidor virtual que permita la ejecución de servicios de base de datos y servicios web.
- Medio de almacenamiento extraíble con al menos 8 GB de espacio libre.

5.6. DESARROLLO DEL COMPONENTE

5.6.1. Etapas de desarrollo

El desarrollo del componente de software se realizó por iteraciones, según lo estipulado en la metodología *XP*, en donde cada una de ellas entregaba un prototipo del componente. El desarrollo de funcionalidades se organizó de la siguiente manera:

- **Iteración 1:** en esta iteración se planteó el esquema básico para las páginas HTML que fueron usadas en el proyecto, incluyendo una primera versión de algunos formularios.
- **Iteración 2:** se trabaja sobre el clasificador ingenuo de Bayes, al finalizar la iteración se obtiene el reporte de clasificación y la matriz de confusión sin formato.
- **Iteración 3:** se trabaja sobre el clasificador árbol de decisión, se obtienen los mismos resultados que con el clasificador ingenuo de Bayes.

- **Iteración 4:** se integró una herramienta de validación de código fuente sobre el repositorio, y se realizó corrección de errores de estilo en el código.
- **Iteración 5:** se trabaja sobre el clasificador máquinas de soporte vectorial, aún no se integra el formato a los resultados obtenidos.
- **Iteración 6:** se trabaja sobre el clasificador k-nearest neighbors, y corrección de errores en el código.
- **Iteración 7:** se da formato a la salida de los procesos de entrenamiento y se trabaja sobre el clasificador red neuronal artificial.
- **Iteración 8:** se agrega la afinación de los parametros de los clasificadores y se trabaja sobre la funcionalidad de clasificación de datasets.

5.6.2. Pruebas

Pruebas al componente de software

Antes de finalizar cada iteración se realizaban pruebas de funcionalidad básicas (sobre la funcionalidad que se estaba desarrollando) y en las iteraciones de corrección de errores se realizaron pruebas de funcionalidad completas (sobre todo el desarrollo).

Para verificar la calidad del código fuente, se realizó la integración de una herramienta para el análisis automático de código al repositorio de GitHub del componente. Haciendo uso de esta herramienta se realizó el seguimiento de problemas relacionados a la calidad del código fuente, y corrección de los mismos.

Debido al tiempo que se tenía para el desarrollo del proyecto no se realizaron pruebas unitarias, de carga o de estrés sobre el componente, las pruebas aplicadas fueron exclusivamente a nivel de desarrollador, en las cuales se buscó el aseguramiento de la funcionalidad desarrollada.

Pruebas a los algoritmos

En cuanto a los algoritmos de aprendizaje supervisado utilizados en el proyecto[1] se realizaron pruebas sobre el rendimiento de los mismos en la etapa de entrenamiento, para esto se definió la siguiente secuencia.

1. Etapa de pre procesamiento.
   a) En esta etapa se realizó la carga del dataset, una vez cargado dentro del componente, se realiza la eliminación de las filas con valores nulos dentro de la columna de datos.
   b) Posterior a eso se realiza la eliminación de los signos de puntuación y conectores, gracias a la implementación de la librería NLTK[2].
   c) Luego de esto se se realiza un proceso de extracción de características y se generan unigramas y bigramas sobre los datos de entrada.

2. División del dataset.
   a) Para poder verificar la calidad de los resultados de los clasificadores, se realiza la división del dataset de entrenamiento de la siguiente forma: 70 % de los datos se destinan para el entrenamiento del dataset y el 30 % restante se usa para validar el entrenamiento.
   b) Al terminar se verifica que el dataset se encuentre balanceado correctamente entre las clases de entrada, de no estarlo se procede a realizar un submuestreo aleatorio buscando evitar el sobre entrenamiento de los algoritmos.

3. Entrenamiento del algoritmo: Con los datos correctamente balanceados entre las clases de entrada, se procede a entrenar el algoritmo seleccionado.

4. Verificación del algoritmo: Con el 30 % restante de los datos de entrada, se realiza una extracción de características básica para dejar los datos de entrada de manera que coincida con el vocabulario de entrenamiento.

5. Recolección de datos: Al finalizar la etapa de verificación, se extraen las métricas y matriz de confusión para cada uno de los algoritmos que se haya entrenado. En la imagen 5-5 y 5-6 se muestran un ejemplo sobre el reporte que entregan los algoritmos y la matriz de confusión para el algoritmo árbol de decisión.

[1] F. Pedregosa y col. "Scikit-learn: Machine Learning in Python". En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

[2] Steven Bird, Ewan Klein y Edward Loper. *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc., 2009. ISBN: 0596516495, 9780596516499.

Figura 5-5.: Captura de pantalla del reporte de clasificación para el algoritmo árbol de decisión

Figura 5-6.: Caputa de pantalla de la matriz de confusión para el algoritmo árbol de decisión

Cuando se están usando los algoritmos ya entrenados y verificados de aprendizaje supervisado en tareas de clasificación, se realiza la medición del tiempo que le tomó a cada algoritmo seleccionado clasificar el mismo dataset y se gráfica la comparación, tal y como se muestra

en la figura 5-7, en donde se comparan el tiempo empleado por los clasificadores máquinas de soporte vectorial y el clasificador ingenuo de Bayes.

Figura 5-7.: Ejemplo de gráfica resultado para el proceso de clasificación

6. RESULTADOS

Dentro de la implementación del componente de software se generaron los manuales de instalación y usuario para facilitar la interacción e instalación del componente de software, estos pueden ser consultados en los anexos del presente documento.

En las pruebas realizadas, se obtuvieron resultados satisfactorios en cuanto al tiempo de respuesta de los algoritmos logrando clasificar aproximadamente 100 elementos en menos de un segundo.

De acuerdo con los resultados observados durante la caracterización del estado del arte, el clasificador que presentó mejores resultados fue el clasificador máquinas de soporte vectorial, el cual dio como resultado el 84 % de precisión en promedio en las tareas de clasificación.

Por su parte se identificó que el algoritmo que entregó peores resultados fue el árbol de decisión con un promedio del 76 %.

Sin embargo durante el proceso de desarrollo se pudo identificar que variaciones sobre la calidad del dataset afectan significativamente los resultados entregados por los clasificadores, lo que demuestra una correlación entre los datos y los resultados obtenidos para el caso de los algoritmos de aprendizaje supervisados. Se detectó también que es importante lograr separar las características de las clases objetivo, para mejorar la calidad de los resultados, ya que según se pudo ver en casos donde las características no estaban diferenciadas fuertemente entre dos clases, se pueden llegar a presentar elementos falsos positivos.

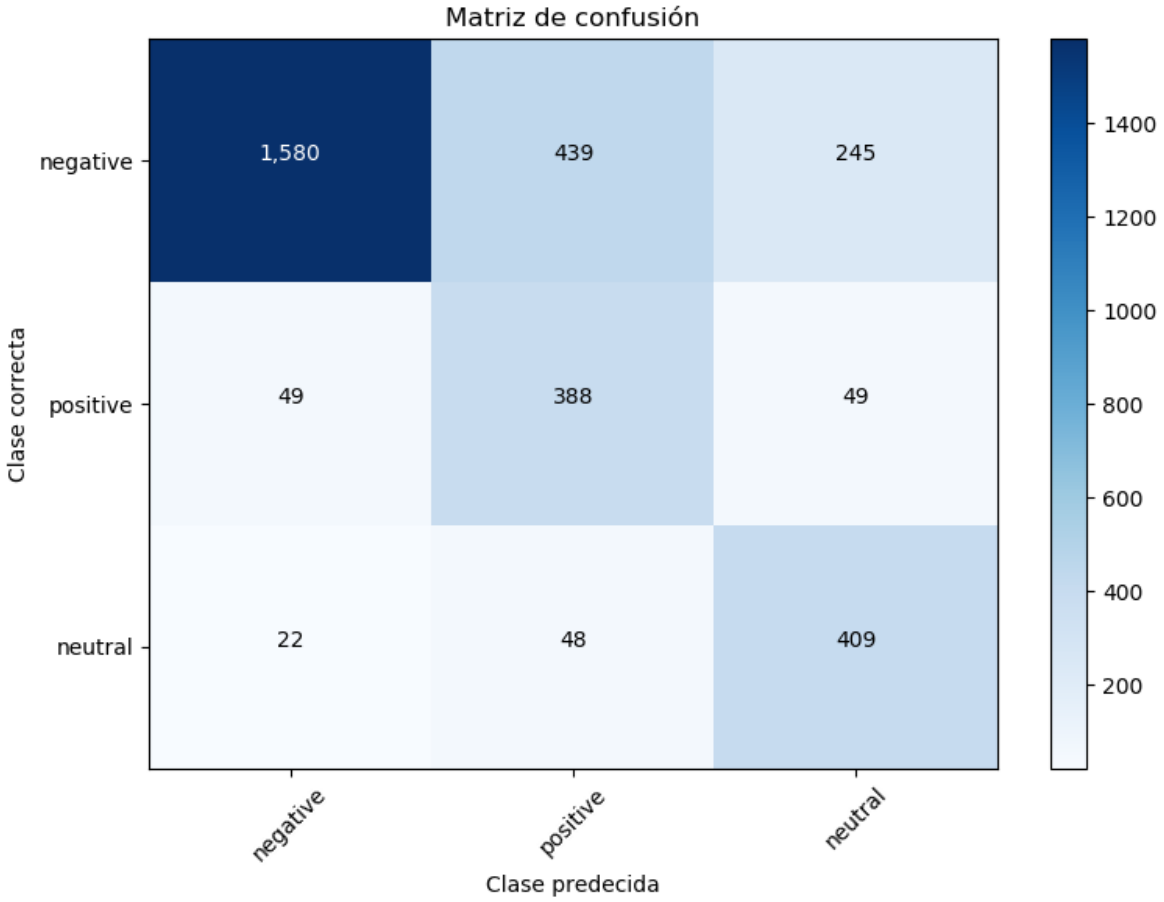
A continuación se presentan los resultados obtenidos para cada uno de los clasificadores, junto a la matriz de confusión generada en cada caso.

- Clasificador de Bayes ingenuo: en la tabla 6-1 se muestra el resumen de la etapa de entrenamiento, y en la imagen 6-1 la matriz de confusión

Tabla 6-1.: Resultados para el clasificador ingenuo de Bayes

precisión	recall	f1	Cantidad de datos
0.82	0.74	0.75	3229

Figura 6-1.: Matriz de confusión para el clasificador ingenuo de Bayes

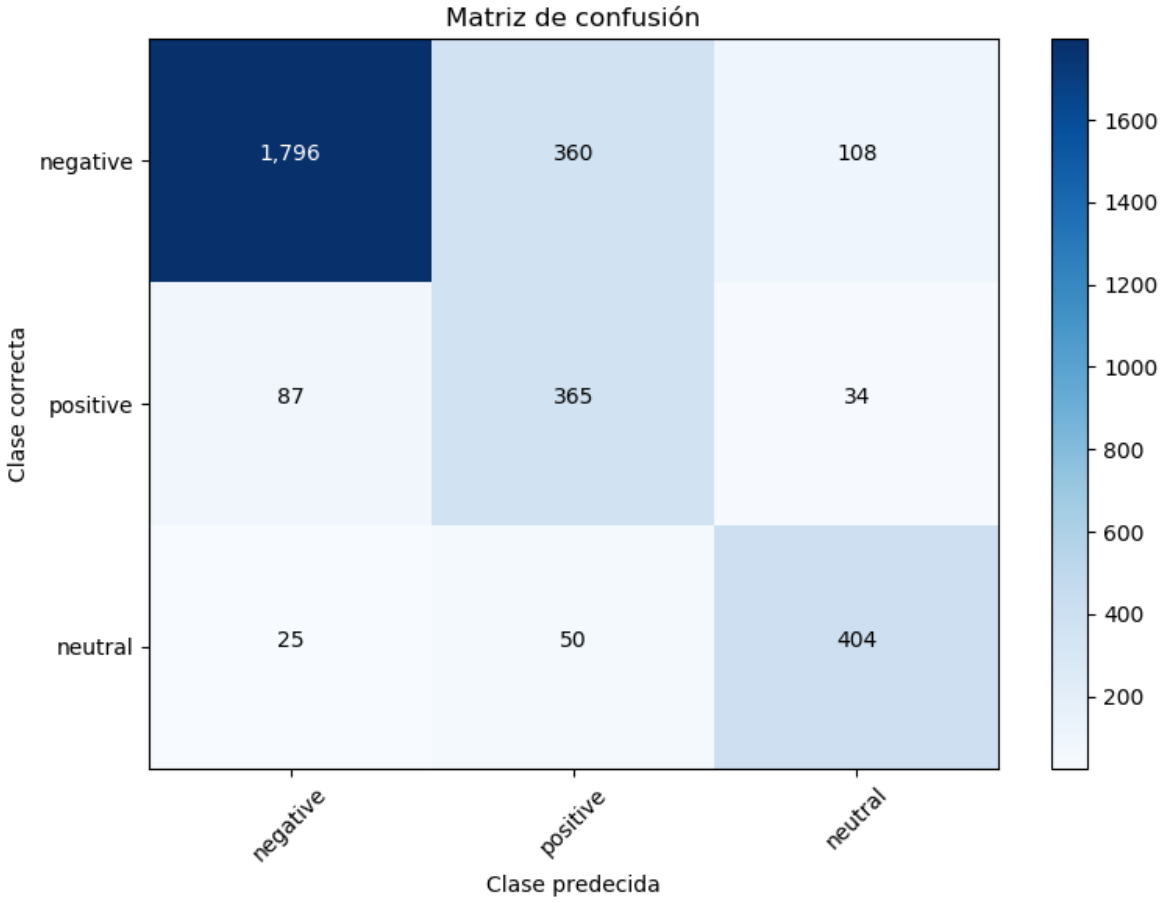


- Máquinas de soporte vectorial: en la tabla 6-2 se muestra el resumen de la etapa de entrenamiento, y en la imagen 6-2 la matriz de confusión

Tabla 6-2.: Resultados para el clasificador máquinas de soporte vectorial

precisión	recall	f1	Cantidad de datos
0.84	0.79	0.81	3229

Figura 6-2.: Matriz de confusión para el clasificador máquinas de soporte vectorial



- Árboles de decisión: en la tabla 6-3 se muestra el resumen de la etapa de entrenamiento, y en la imagen 6-3 la matriz de confusión

Tabla 6-3.: Resultados para el clasificador árbol de decisión

precisión	recall	f1	Cantidad de datos
0.76	0.63	0.66	3229

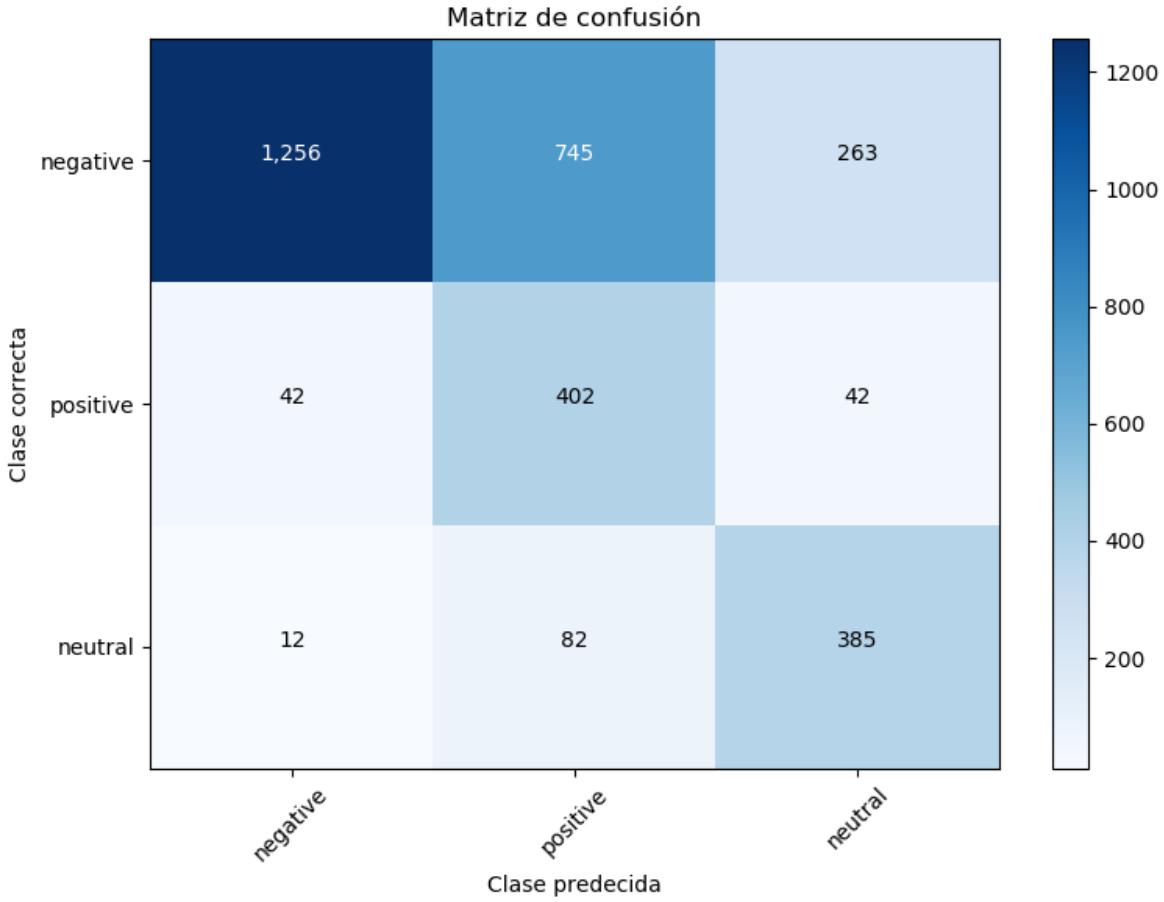
Figura 6-3.: Matriz de confusión para el clasificador árbol de decisión

- K-Nearest Neighbors: en la tabla 6-4 se muestra el resumen de la etapa de entrenamiento, y en la imagen 6-4 la matriz de confusión

Tabla 6-4.: Resultados para el clasificador K-Nearest Neighbors

precisión	recall	f1	Cantidad de datos
0.80	0.63	0.66	3229

Figura 6-4.: Matriz de confusión para el clasificador K-Nearest Neighbors

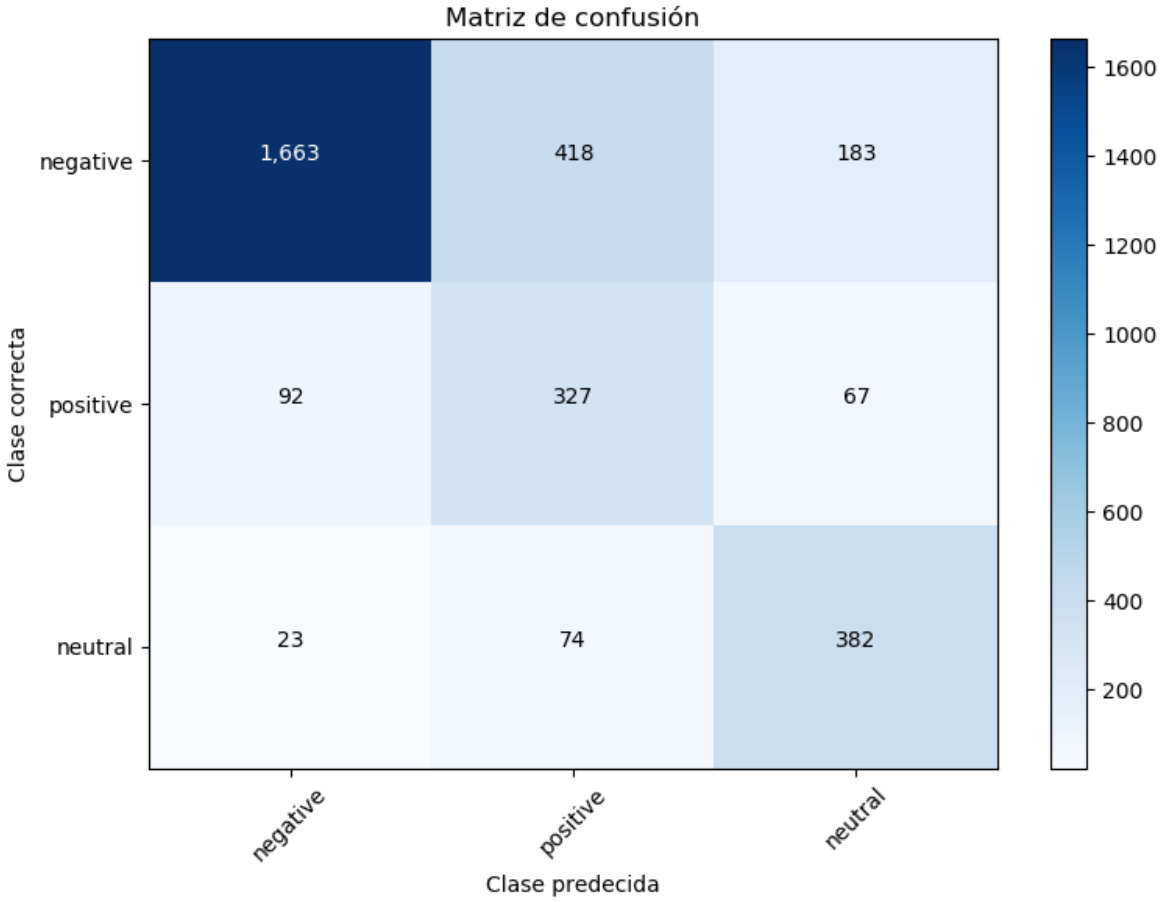


- Redes neuronales artificiales: en la tabla 6-5 se muestra el resumen de la etapa de entrenamiento, y en la imagen 6-5 la matriz de confusión

Tabla 6-5.: Resultados para el clasificador Redes neuronales artificiales

precisión	recall	f1	Cantidad de datos
0.81	0.73	0.75	3229

Figura 6-5.: Matriz de confusión para el clasificador Redes neuronales artificiales



7. CONCLUSIONES

1. Los algoritmos tienden a mejorar las respuestas que dan con respecto a los parámetros con los cuales se configuren, sin embargo se pueden presentar problemas al usar determinadas combinaciones de parámetros, por esto que se permita la parametrización completa de los algoritmos por parte del usuario final, puede acarrear pérdidas en la calidad de las predicciones realizadas, y fallas en el funcionamiento de la herramienta, en este aspecto se encontró que es mejor plantear una parametrización "general" que favorezca un desempeño estándar de los algoritmos, mediante el uso de evaluadores provistos por la librería de aprendizaje automático usada.

2. El componente de software desarrollado durante la realización de este proyecto es capaz de abordar problemas de clasificación de textos cortos que no estén directamente relacionados con el objetivo definido por el macro proyecto institucional con el cual se espera integrar, ya que en el proceso de entrenamiento se requieren las clases de los elementos, y basado en estas se realiza el proceso de extracción de características, lo que le da versatilidad al componente y lo hace fácilmente integrable con otras plataformas.

3. La caracterización del estado del arte permitió encontrar fuertes similitudes en gran parte de la literatura relacionada con la clasificación de textos cortos, en esta, se hace también evidente que la etapa de pre procesamiento de los datos, es fundamental y determinante en el proceso de entrenamiento de los algoritmos.

4. La realización de este trabajo permitió constatar la aplicabilidad del concepto de clasificación de textos cortos a información extraída de redes sociales, no solamente con el fin de clasificar las tendencias discursivas sino de realizar monitoreo sobre poblaciones que podrían requerir atención especial, mejorando así los mecanismos de reacción que podrían tener entidades de control y prevención.

8. TRABAJO FUTURO

Dentro de los posibles trabajos futuros se recomendaría realizar la revisión sobre los puntos mencionados a continuación:

1. Analizar técnicas que permitan la clasificación de los textos en tiempo real: Es importante según lo descrito en el estado del arte, trabajar en algoritmos que permitan realizar la clasificación de textos cortos en tiempo real, esto debido a que dentro de las aplicaciones del concepto se encuentra el monitoreo de sentimientos que pueden conllevar a problemas de salud pública, por lo tanto se puede mejorar la calidad de este tipo de monitoreo de llegar a permitir la clasificación en tiempo real.

2. Clasificación de texto: Se podría plantear que el componente de software se amplíe en su alcance para permitir la clasificación de textos con longitudes superiores, de esta manera evaluar el impacto en los tiempos de respuesta, y la precisión de las predicciones realizadas por el componente actualmente.

3. Algoritmos de aprendizaje por refuerzo y aprendizaje no supervisado: Si bien el componente está de manera inicial planteado para trabajar con algoritmos de clasificación de textos cortos de manera supervisada, se podría realizar la comparación entre los resultados obtenidos en este estudio haciendo uso de otros algoritmos.

Bibliografía

[1] Charu C. Aggarwal y Chengxiang Zhai. *Mining Text Data*. Vol. 8. New York: Springer Science & Business Media, 2012, pág. 524. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/978-1-4614-3223-4.

[2] Mubashir Ali y col. "A probabilistic framework for short text classification". En: *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. Ene. de 2018, págs. 742-747. DOI: 10.1109/CCWC.2018.8301712.

[3] Armin Ronacher. *Foreword — Flask Documentation (0.12)*. 2010. URL: http://flask.pocoo.org/docs/0.12/foreword/ (visitado 15-04-2018).

[4] Kent Beck y col. *Manifiesto por el Desarrollo Ágil de Software*. URL: http://agilemanifesto.org/iso/es/manifesto.html (visitado 06-08-2017).

[5] Steven Bird, Ewan Klein y Edward Loper. *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc., 2009. ISBN: 0596516495, 9780596516499.

[6] Jason Brownlee. *A Gentle Introduction to the Bag-of-Words Model*. 2017. URL: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (visitado 21-04-2018).

[7] Valentina Jaramillo Bustamante. "Conflicto armado en Colombia, el proceso de paz y la Corte Penal Internacional: Un estudio sobre la internacionalización del conflicto armado en Colombia y su búsqueda por encontrar la paz duradera". En: *Journal of International Law* 6.6 (2015), págs. 6-34. ISSN: 2216-0965.

[8] Cambridge University Press. *Definition of framework*. 2018. URL: https://dictionary.cambridge.org/es/diccionario/ingles/framework (visitado 15-04-2018).

[9] Centro Nacional de Memoria Histórica. *Estadísticas - ¡Basta ya! Colombia: Memorias de guerra y dignidad.* 2016. URL: http://www.centrodememoriahistorica.gov.co/micrositios/informeGeneral/estadisticas.html (visitado 28-03-2017).

[10] Xuetong Chen y col. "What About Mood Swings: Identifying Depression on Twitter with Temporal Measures of Emotions". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1653-1660. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191624. URL: https://doi.org/10.1145/3184558.3191624.

[11] "DECRETO 1377 DE 2013". En: *Diario Oficial 48834 del 27 de junio de 2013* (2013). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=53646%7B%5C#%7D0.

[12] Bart Desmet y Véronique Hoste. "Online suicide prevention through optimised text classification". En: *Information Sciences* 439-440 (2018), págs. 61-78. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.02.014. URL: http://www.sciencedirect.com/science/article/pii/S002002551830094X.

[13] Giorgio Maria Di Nunzio, Maria Maistro y Federica Vezzani. "A Gamified Approach to NaiVe Bayes Classification: A Case Study for Newswires and Systematic Medical Reviews". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1139-1146. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191547. URL: https://doi.org/10.1145/3184558.3191547.

[14] Christiane Fellbaum. *WordNet — A Lexical Database for English*. 2005. URL: https://wordnet.princeton.edu/ (visitado 15-04-2018).

[15] Álvaro Galván Lucas y Jose Manuel Torres Púa. *Extreme programming*. URL: http://osl2.uca.es/wikiCE/index.php/Extreme%7B%5C_%7Dprogramming (visitado 09-08-2017).

[16] Samujjwal Ghosh y Maunendra Sankar Desarkar. "Class Specific TF-IDF Boosting for Short-text Classification: Application to Short-texts Generated During Disasters". En: *Companion of the The Web Conference 2018 on The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, págs. 1629-1637. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191621. URL: https://dl.acm.org/citation.cfm?id=3191621.

[17] Fabio A. González. *Introducción Aprendizaje de Máquina*. 2007. URL: http://dis.unal.edu.co/profesores/fgonza/courses/2007-I/ml/ml-01-introduction.pdf (visitado 11-06-2017).

[18] William L. (Encyclopædia Britannica) Hosch. *machine learning*. 2009. URL: https://global.britannica.com/technology/machine-learning (visitado 26-03-2017).

[19] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). *¿Qué es un blog?* URL: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo%7B%5C_%7D1%7B%5C_%7DIniciacionblog/qu%7B%5C_%7Des%7B%5C_%7Dun%7B%5C_%7Dblog.html (visitado 11-06-2017).

[20] Jason Brownlee. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. 2014. URL: http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (visitado 03-06-2017).

[21] Jason Brownlee. *Supervised and Unsupervised Machine Learning Algorithms*. 2016. URL: http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (visitado 06-06-2017).

[22] José Joskowicz. "Reglas y Prácticas en eXtreme Programming". En: (2008). URL: https://iie.fing.edu.uy/%7B~%7Djosej/docs/XP%20-%20Jose%20Joskowicz.pdf.

[23] Julián Pérez Porto y Ana Gardey. *Definición de Twitter*. 2014. URL: http://definicion.de/twitter/ (visitado 11-06-2017).

[24] Faris Kateb y Jugal Kalita. "Classifying Short Text in Social Media: Twitter as Case Study". En: *International Journal of Computer Applications* 111.9 (2015), págs. 1-12. ISSN: 09758887. DOI: 10.5120/19563-1321. URL: http://research.ijcaonline.org/volume111/number9/pxc3901321.pdf.

[25] H M Keerthi Kumar y col. "Classification of Sentiments in Short-text: An Approach Using mSMTP Measure". En: *Proceedings of the 2Nd International Conference on Machine Learning and Soft Computing*. ICMLSC '18. Phu Quoc Island, Viet Nam: ACM, 2018, págs. 145-150. ISBN: 978-1-4503-6336-5. DOI: 10.1145/3184066.3184074. URL: http://doi.acm.org/10.1145/3184066.3184074.

[26] Patricio Letelier y Mª Carmen Penadés. "Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". En: (). URL: http://users.dsic.upv.es/asignaturas/eui/lds/doc/masyxp.pdf.

[27] "LEY 1273 DE 2009". En: *Diario Oficial 47.223 de enero 5 de 2009* (2009). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492.

[28] "LEY 1712 DE 2014". En: *Diario Oficial 49084 de marzo 6 de 2014* (2014). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=56882.

[29] "LEY ESTATUTARIA 1581 DE 2012". En: *Diario Oficial 48587 de octubre 18 de 2012.* (2012). URL: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=49981.

[30] Vishal Maini y Samir Sabri. *Reinforcement Learning*. 2017. URL: https://medium.com/machine-learning-for-humans/reinforcement-learning-6eacf258b265 (visitado 14-04-2018).

[31] Austin Mccartney. *"How Short is a Piece of String?": An Investigation into the Impact of Text Length on Short-Text Classification Accuracy*. 2017.

[32] Walaa Medhat, Ahmed Hassan y Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". En: *Ain Shams Engineering Journal* 5.4 (2014), págs. 1093-1113. ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2014.04.011. URL: http://www.sciencedirect.com/science/article/pii/S2090447914000550.

[33] Tom M Mitchell. *Machine Learning*. Vol. 4. 1997, págs. 417-433. ISBN: 9781577354260. DOI: 10.1145/242224.242229. arXiv: 0-387-31073-8.

[34] Tom M Mitchell. "The Discipline of Machine Learning". En: *Machine Learning* 17.July (2006), págs. 1-7. ISSN: 0264-0414. DOI: 10.1080/026404199365326. arXiv: 9605103 [cs]. URL: http://www-cgi.cs.cmu.edu/%7B~%7Dtom/pubs/MachineLearningTR.pdf.

[35] Tania Camila Niño y col. "Uso de redes neuronales artificiales en predicción de morfología mandibular a través de variables craneomaxilares en una vista posteroanterior". En: (2016). DOI: http://dx.doi.org/10.11144/Javeriana.uo35-74.urna.

[36] Yuanxin Ouyang y col. "News Title Classification with Support from Auxiliary Long Texts". En: *Neural Information Processing*. Ed. por Chu Kiong Loo y col. Cham: Springer International Publishing, 2014, págs. 581-588. ISBN: 978-3-319-12640-1.

[37] Oxford University Press. *Definition of bigram*. 2018. URL: https://en.oxforddictionaries.com/definition/bigram (visitado 14-04-2018).

[38] Oxford University Press. *Definition of trigram*. 2018. URL: https://en.oxforddictionaries.com/definition/trigram (visitado 14-04-2018).

[39] F. Pedregosa y col. "Scikit-learn: Machine Learning in Python". En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

[40] Real Academia Española. *DLE: lematizar*. 2014. URL: http://dle.rae.es/?id=N6LviQx (visitado 15-04-2018).

[41] República de Colombia. Corte Constitucional. "Constitución Política de Colombia". En: Normatividad 5 - 2015 (1991), pág. 125. URL: http://www.corteconstitucional.gov.co/inicio/Constitucion%20politica%20de%20Colombia%20-%202015.pdf.

[42] Irina Rish. "IBM Research Report An empirical study of the naive Bayes classifier". En: (2001). URL: http://domino.watson.ibm.com/library/CyberDig.nsf/home.

[43] Tajinder Singh y Madhu Kumari. "Role of Text Pre-processing in Twitter Sentiment Analysis". En: *Procedia Computer Science* 89 (2016). Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India, págs. 549-554. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.06.095. URL: http://www.sciencedirect.com/science/article/pii/S1877050916311607.

[44] Ge Song y col. "Short Text Classification: A Survey". En: *Journal of Multimedia* 9.5 (2014), págs. 635-643. ISSN: 1796-2048. DOI: 10.4304/jmm.9.5.635-643. URL: http://ojs.academypublisher.com/index.php/jmm/article/view/12635.

[45] Twitter International Company. *Política de Privacidad de Twitter*. 2017. URL: https://twitter.com/privacy?lang=es%7B%5C#%7Dupdate (visitado 11-06-2017).

[46] Twitter International Company. *Terminos del Servicio*. 2016. URL: https://twitter.com/tos?lang=es (visitado 11-06-2017).

[47] Fang Wang y col. "Concept-based Short Text Classification and Ranking". En: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14* (2014), págs. 1069-1078. DOI: 10.1145/2661829.2662067. URL: http://dl.acm.org/citation.cfm?doid=2661829.2662067.

[48] Don Wells. *Extreme Programming Rules*. URL: http://www.extremeprogramming.org/rules.html (visitado 09-08-2017).

[49] Jian Xu y col. "Signature based trouble ticket classification". En: *Future Generation Computer Systems* 78 (2018), págs. 41-58. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.07.054. URL: http://www.sciencedirect.com/science/article/pii/S0167739X16308056.

[50] Rui Xu. "Survey of clustering algorithms for MANET". En: *IEEE Transactions on Neural Networks* 16.3 (2005), págs. 645-678. ISSN: 1045-9227. DOI: 10.1109/TNN.2005.845141. arXiv: 0912.2303. URL: http://arxiv.org/abs/0912.2303.

A. ANEXO: SRS-1804-Componente de clasificación

En este anexo se realiza la especificación de requerimientos para el componente de software desarrollado en este proyecto

Componente de Clasificación de Textos Cortos Basado en Técnicas de Aprendizaje Automático

ESPECIFICACIÓN DE REQUERIMIENTOS

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Abril, 2018	Juan Jiménez	1.0	Descripción del documento y definición inicial de requerimientos funcionales del componente.
Abril, 2018	Juan Jiménez	2.0	Definición de requerimientos no funcionales.

Introducción

Resumen

Los procesos de clasificación de texto, constituyen una de las áreas de estudio del aprendizaje automático o aprendizaje de máquina. El aprendizaje de máquina (*Machine Learning*) es una disciplina de la Inteligencia Artificial encargada del diseño de algoritmos que permiten a las computadoras encontrar patrones a partir de un conjunto de datos y lograr aprender de los mismos [1], [2].

El presente documento es una descripción completa del comportamiento de un componente de analítica para la clasificación de textos cortos basado en técnicas de aprendizaje automático. Esta descripción incluye un conjunto de requerimientos funcionales mediante los cuales se busca describir todas las interacciones que tendrán los usuarios con el software. También se describen los requerimientos no funcionales del software.

Propósito

Identificar las necesidades que se tienen en el macro proyecto “Efectividad de un protocolo de re experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política” y basados en estas especificar los requerimientos funcionales y no funcionales que permitan desarrollar un componente de analítica para la clasificación de textos cortos, que se pueda integrar como parte de la plataforma propuesta como solución al mismo.

Audiencia

Este documento en primera instancia se dirige al equipo de investigación que trabaje dentro del proyecto “Efectividad de un protocolo de re experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política” y al equipo encargado de desarrollar, mantener y/o actualizar el componente de software, con el objeto de establecer una hoja de ruta para el desarrollo, las pruebas a realizar y la implementación del producto de software.

Autores

Nombre	Raúl Ernesto Menéndez Mora
Rol	Gestor del proyecto
Categoría Profesional	Ingeniero de Sistemas
Contacto	remenendez@ucatolica.edu.co

Nombre	Juan Sebastián Jiménez Valero
Rol	Analista de Software
Categoría Profesional	Estudiante de ingeniería de sistemas
Contacto	jsjimenez50@ucatolica.edu.co

Ámbito del Sistema

Contexto: El componente debe integrarse como parte del framework propuesto para el macro proyecto: "*Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política*". Inicialmente está orientado hacia la clasificación de textos cortos y las entradas serán tweets o comentarios en facebook, mientras las salidas serán categorías definidas previamente.

Función y rendimiento: El componente de clasificación deberá tener la opción de poder entrenarse con un conjunto de datos de entrada, una vez terminada esta fase de entrenamiento, el software podrá recibir textos cortos como entrada y devolverá la clasificación de dicho texto teniendo en cuenta las categorías definidas en la etapa de entrenamiento. La respuesta a la clasificación de un texto corto deberá realizarse en menos de 5 segundos.

Definiciones, Acrónimos y Abreviaturas

Definiciones

Datasets: Conjunto de datos.

Machine learning: El aprendizaje de máquina, aprendizaje automático o machine learning, busca estudiar la construcción de sistemas, que vayan mejorando según la experiencia adquirida mediante el tratamiento del problema que se está tratando. Se puede decir, que un sistema aprende de una experiencia E, si su desempeño en la realización de la tarea T, con base a la medida de desempeño P muestra mejoría con respecto a la variable de desempeño bajo la cual se está midiendo después de varias sesiones de trabajo resolviendo el problema [3], [4].

Para ejemplificar lo anterior decimos que si tenemos un algoritmo cuyo objetivo es jugar a las damas, la manera en la cual podríamos relacionarlo con el esquema mencionado, sería así:

- Se dice que la tarea T es jugar a las damas.
- La medida de desempeño P será el porcentaje de juegos ganados contra oponentes.
- La experiencia de entrenamiento E por su parte será jugar partidas de práctica contra sí mismo

El aprendizaje de máquina se puede clasificar según el tipo de aprendizaje que apliquen en [5]:

1. **Aprendizaje supervisado:** Se parte de un conjunto de datos del cual se conoce la salida Yn correspondiente a la entrada Xn, a este tipo de datos se le conoce como datos etiquetados. Busca generalizar un modelo que permita predecir para datos de entrada nuevos su correspondiente salida.

2. **Aprendizaje no supervisado:** En este modelo de aprendizaje sólo se cuenta con el dato de entrada Xn, o sea, datos no etiquetados. Se busca encontrar distribuciones en los datos de entrada, o patrones relacionados a los mismos.

3. **Aprendizaje semi-supervisado:** Es una clase de técnicas de aprendizaje automático que utiliza datos de entrenamiento tanto etiquetados como no etiquetados, normalmente una pequeña cantidad de datos etiquetados junto a una gran cantidad de datos no etiquetados [6].

4. **Aprendizaje por refuerzo:** Es una clase de técnicas de aprendizaje automático que utiliza recompensas y penalizaciones que son aplicados al algoritmo buscando que se obtenga la mejor solución posible al problema que se esté analizando [7].

Ingeniería de rasgos: La ingeniería de rasgos busca solucionar el problema de obtener la mayor cantidad de características aprovechables a partir de los datos de entrada, ya que esto influye directamente en el éxito del módelo predictivo usado en la solución de la problemática estudiada [8].

Clasificación de textos: La clasificación de textos, constituye una categoría de algoritmos supervisados, donde la variable de salida es conocida. Éstos algoritmos buscan agrupar una cantidad finita de datos en un conjunto también finito de grupos o categorías. En este caso se entrena al sistema para aprender cómo debe realizar la clasificación de los datos de entrada según una serie de conjuntos de datos de entrenamiento, los cuales están previamente clasificados por un experto, de allí se genera un modelo de clasificación que relaciona las características del dato que se está analizando con una de las categorías de destino [9].

Clasificación de textos cortos: Es un subconjunto dentro de la categoría de la clasificación de textos. Para este subconjunto del problema de clasificación se debe hacer frente a ciertas dificultades claves para el proceso las cuales son la longitud de los textos disponibles, y la cantidad de textos que puedan estar relacionadas entre sí. Esto conlleva a aplicar un proceso de limpieza de la información y una pre clasificación donde se separen los datos que puedan ser útiles de los que no lo sean, por lo tanto se presenta un aumento en el tiempo previo al procesamiento de la información útil, lo cual aumenta la dificultad para realizar análisis en tiempo real [10].

Blog: Es un sitio web que se actualiza constantemente y en donde el autor puede publicar información que sea de su interés, generalmente las publicaciones del sitio se organizan en forma cronológica inversa, es decir primero se mostrarán las publicaciones o entradas más recientes [11].

Datos Abiertos: Son todos aquellos datos primarios o sin procesar, que se encuentran en formatos estándar e interoperables que facilitan su acceso y reutilización, los cuales están bajo la custodia de las entidades públicas o privadas que cumplen con funciones públicas y que son puestos a disposición de cualquier ciudadano, de forma libre y sin restricciones, con el fin de que terceros puedan reutilizarlos y crear servicios derivados de los mismos [12].

Microblogging: Es una variante a los blogs que se carácteriza porque sus entradas son generalmente más cortas [13].

Texto corto: Se considera un texto corto a uno cuya extensión se encuentre entre los 30 y los 400 caracteres, generalmente son utilizados en publicaciones de redes sociales, mensajes de chat, o mensajes de texto; Se caracterizan por su carácter de comunicación inmediata por lo que cuentan con pocas palabras y características aprovechables, en muchas ocasiones se usan términos del argot [14].

Tuit: El término Tuit o en inglés Tweet son todas aquellas publicaciones que pueden enviar los usuarios registrados en Twitter, las cuales constan de un máximo de 280 caracteres, los tuits

pueden incluír imágenes, vídeos o enlaces a sitios web. De forma predeterminada estos son públicos [15].

Twitter: Twitter es una red social de microblogging originada en el año 2006 que permite realizar actualizaciones en tiempo real, se caracteriza por la longitud de sus entradas, más conocidas como tuits [13].

Árboles de decisión (DT): Un árbol de decisión es una descomposición jerárquica que se realiza sobre el dataset de pruebas, en donde se busca que un predicado o una condición dividan los datos. En el ámbito de la clasificación de textos, estos predicados atienden a la presencia o ausencia de ciertas palabras en el texto, lo cual dota de sentido al mismo. En este tipo de clasificadores luego de generar una estructura arbórea, se utilizarán datos que son parte del conjunto de entrenamiento, y que no fueron usados para la generación de la estructura, para verificar si esta es lo suficientemente granular para solucionar el problema; ya que se pondrán a prueba los datos y de poder segmentarlos aún más en nuevas subcategorías (Hojas) se deberá dividir nuevamente la rama del árbol para lograr que la clasificación sea lo más detallada posible [9].

Clasificadores basados en reglas: Estos clasificadores establecen un concepto similar a los árboles de decisión ya que estos, se rigen por un conjunto de reglas que condicionan la pertenencia a una clase esperada. Por lo general, estas reglas validan la existencia de ciertos predicados dentro del texto, más no su ausencia como lo hacen los árboles de decisión; sin embargo, toda una rama de un árbol se puede expresar como una regla para un clasificador de este tipo. Generalmente las reglas son condiciones sencillas y representan un conjunto de términos que deben estar presentes en el texto para cumplir con la condición planteada de pertenencia a una clase. El objetivo principal de este tipo de clasificadores es generar un conjunto de reglas que cubra todos los posibles predicados que puedan estar en el texto a clasificar al menos por una regla [9].

Máquinas de soporte vectorial (SVM): Estas basan su funcionamiento en la determinación de diferencias entre las clases que se tienen disponibles para la clasificación de los datos de entrada. Estas diferencias trazan límites de pertenencia a las clases, ya que los textos de una misma clase guardan cierta relación entre si, lo cual brinda soporte a las tareas de clasificación en donde se determinará en base a estas diferencias si el dato pertenece o no a determinada clase [9].

Redes Neuronales Artificiales: Las redes neuronales se relacionan con las máquinas de soporte vectorial ya que ambos modelos buscan realizar las tareas de clasificación a partir de las diferencias que separan una clase de la otra, para este caso las redes neuronales artificiales (ANN, del inglés Artificial Neural Network), se compone de varias unidades o “neuronas”, las cuales, a partir de un cojunto de datos de entrada, brindan una salida enmarcada dentro de estas diferencias. La clasificación que se realiza dentro de una red neuronal se basa en el peso que tengan las palabras contenidas en el texto estudiado. Los pesos en la red son modificados con el fin de reducir el margen de error en la tarea realizada, en este tipo de clasificadores se suelen usar varias capas de neuronas que son alimentadas por el proceso realizado en las capas inferiores [16]; el objetivo de tener varias

capas es mejorar la precisión en la selección de la clase de salida mediante el análisis de varios límites de la misma [9].

Clasificadores Bayesianos: Dentro de los clasificadores probabilísticos se encuentran los clasificadores ingenuos (del inglés, naive) de Bayes; estos son ampliamente usados a pesar de su simplicidad, ya que su base teórica entrega resultados sobresalientes para varias tareas de clasificación. A estos clasificadores se les llama ingenuos porque modelan la distribución de los términos presentes en los textos trabajando bajo la presunción de que la distribución de diferentes términos (es decir su presencia o ausencia) es independiente de la de otros. Esto mejora la calidad de los resultados obtenidos al trabajar en ambientes con variables "ruidosas" como la clasificación de textos, donde no todas las palabras contenidas en el texto a estudiar contienen significado, o aportan a la tarea de clasificación [9], [17].

Acrónimos, Abreviaturas

ANN: Redes neuronales artificiales, Artificial Neural Network por sus siglas en inglés.

DT: Árbol de decisión, Decision Tree por sus siglas en inglés.

REST: Es un estilo arquitectural orientada a servicios distribuidos en red, las siglas REST vienen del inglés *REpresentational State Transfer* [18].

SVM: Máquina de soporte vectorial, Support Vector Machine por sus siglas en inlgés.

Visión General del Documento

Alcance

1. Se deberá entregar un componente de software, que permitirá clasificar los datos de entrada en una categoría psicológica conocida.

2. El componente deberá permitir su integración con el framework propuesto para abordar la solución a la problemática planteada en el proyecto "Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política".

Limitaciones

1. La infraestructura disponible para los procesos involucrados con el desarrollo del producto es básica y limitada.

Descripción

Funciones del producto

El componente permitirá la carga de conjuntos de datos, definición de las categorías o clases de salida, entrenar un clasificador usando un conjunto de datos cargado, realizar tareas de clasificación sobre conjuntos de datos usando algoritmos de aprendizaje supervisado que hayan sido previamente entrenados, representar gráficamente el rendimiento de un clasificador previamente entrenado.

Características de los Usuarios

El componente está pensado para funcionar como parte de un framework mayor o para trabajar independientemente. El consumo del mismo por cualquiera de las dos vías debe realizarse por medio de web-services. La arquitectura general del componente es tipo REST.

Restricciones

- Interfaz para ser usada únicamente en internet.
- Se consumirá el componente como un servicio web que tendrá arquitectura REST.

Suposiciones y Dependencias

Ver requerimientos específicos.

Requisitos futuros

1. Analizar técnicas que permitan la clasificación de los textos en tiempo real.
2. Ampliar el alcance del componente para que se permita la clasificación de textos con longitudes superiores.

Metodología

Durante el desarrollo del componente se deberá priorizar la gestión de riesgos en la realización del proyecto debido a la alta probabilidad de cambios en los requerimientos; de acuerdo a esto se utilizará una metodología ágil que se ajuste a los principios planteados por el manifiesto por el desarrollo ágil de software [19].

La metodología *X.P.* Extreme Programming es una metodología ágil que puede ser aplicada en los casos en que se requiera gestionar un alto riesgo de cambio en los requerimientos mediante la comunicación constante entre todos los participantes del proceso de desarrollo en donde se incluye al cliente como parte activa, y el planteamiento que supone mediante el desarrollo representa un alto valor en los proyectos donde se requiere la agilidad de las soluciones presentadas [20].

Requerimientos Específicos

Requerimientos funcionales

A continuación, se detallan los requisitos que el sistema deberá implementar.

Código	RF_001
Nombre de requerimiento	Cargar conjunto de datos.
Prioridad	Alta
Descripción	Se debe permitir la carga de un conjunto de datos. Los formatos permitidos para la carga de datos son: txt, csv, xls y xlsx.
Entradas	Fichero que contiene el conjunto de datos.
Proceso:	Una vez seleccionado el fichero y ejecutado el proceso de carga, el mismo debe estar disponible para el uso en el componente.
Salidas:	Ninguna

Código	RF_002
Nombre de requerimiento	Definir atributo de salida.
Prioridad	Alta
Descripción	Se debe permitir especificar cuál será el atributo o clase de salida. Este atributo se tendrá en cuenta para realizar la clasificación.
Entradas	El nombre de una columna en un conjunto de datos.
Proceso:	Una vez especificado el nombre de la columna, esta será tomada por cualquier algoritmo de aprendizaje supervisado como la clase objetivo.
Salidas:	Ninguna

Código	RF_003
Nombre de requerimiento	Entrenar clasificador Naive Bayes (NB).
Prioridad	Alta
Descripción	Permite el entrenamiento del clasificador (NB) a partir de un conjunto de datos previamente cargado.
Entradas	Conjunto de datos.
Proceso:	Se debe ejecutar el procedimiento que permite el aprendizaje del clasificador NB.
Salidas:	Devuelve el clasificador entrenado (un fichero) y las métricas precisión, recall y F-measure.

Código	RF_004
Nombre de requerimiento	Entrenar clasificador Árbol de Decisión (DT).
Prioridad	Alta
Descripción	Permite el entrenamiento del clasificador (DT) a partir de un conjunto de datos previamente cargado.
Entradas	Conjunto de datos.
Proceso:	Se debe ejecutar el procedimiento que permite el aprendizaje del clasificador DT.
Salidas:	Devuelve el clasificador entrenado (un fichero) y las métricas precisión, recall y F-measure.

Código	RF_005
Nombre de requerimiento	Entrenar clasificador Máquina de Soporte Vectorial (SVM).
Prioridad	Alta
Descripción	Permite el entrenamiento del clasificador (SVM) a partir de un conjunto de datos previamente cargado.
Entradas	Conjunto de datos.
Proceso:	Se debe ejecutar el procedimiento que permite el aprendizaje del clasificador SVM.
Salidas:	Devuelve el clasificador entrenado (un fichero) y las métricas precisión, recall y F-measure.

Código	RF_006
Nombre de requerimiento	Entrenar clasificador Red Neuronal Artificial (ANN).
Prioridad	Media
Descripción	Permite el entrenamiento del clasificador (ANN) a partir de un conjunto de datos previamente cargado.
Entradas	Conjunto de datos.
Proceso:	Se debe ejecutar el procedimiento que permite el aprendizaje del clasificador ANN.
Salidas:	Devuelve el clasificador entrenado (un fichero) y las métricas precisión, recall y F-measure.

Código	RF_007
Nombre de requerimiento	Entrenar clasificador k-nearest neighbors (KNN).
Prioridad	Alta
Descripción	Permite el entrenamiento del clasificador (KNN) a partir de un conjunto de datos previamente cargado.
Entradas	Conjunto de datos.
Proceso:	Se debe ejecutar el procedimiento que permite el aprendizaje del clasificador K-means.
Salidas:	Devuelve el clasificador entrenado (un fichero) y las métricas precisión, recall y F-measure

Código	RF_008
Nombre de requerimiento	Generar gráficas de rendimiento.
Prioridad	Media
Descripción	Se deben generar gráficos en donde se evidencie el desempeño de cada uno de los algoritmos. Esta gráfica se representará por medio de un fichero png o eps.
Entradas	Clasificador entrenado.
Proceso:	Luego de finalizar un proceso de clasificación generar el respectivo gráfico.
Salidas:	Fichero png con la gráfica del clasificador pasado como entrada.

Requerimientos no funcionales

Código	RNF_001
Nombre	Tiempo de respuesta.
Atributo de calidad	Rendimiento.
Prioridad	Alta
Justificación	El componente de software debe tener una respuesta acorde en sus funcionalidades, de acuerdo al hardware donde esté instalado.
Medida de respuesta	La clasificación de un texto corto se debe realizar en un tiempo no mayor a 5 segundos.

Código	RNF_002
Nombre	Facilidad de uso.
Atributo de calidad	Usabilidad
Prioridad	Alta
Justificación	El componente de software debe ser fácil de aprender a administrar y operar
Medida de respuesta	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 8 horas

Código	RNF_003
Nombre	Documentación del componente.
Atributo de calidad	Usabilidad
Prioridad	Alta
Justificación	Deben existir procedimientos físicos para el manejo del componente de software
Medida de respuesta	El sistema debe contar con manuales de usuario estructurados adecuadamente.

Código	RNF_004
Nombre	Interacción con otros sistemas.
Atributo de calidad	Extensibilidad.
Prioridad	Alta
Justificación	El componente de software debe permitir su fácil integración a otras plataformas web
Medida de respuesta	El componente deberá tener un API que facilite las tareas de integración con otras plataformas

Requerimientos de Licenciamiento

Para el licenciamiento del componente de software se usará una licencia de tipo Creative Commons. Los detalles de la licencia están por definir por el grupo de investigación GISIC.

Bibliografía

[1] T. M. Mitchell, *Machine learning*, vol. 4. 1997.

[2] W. L. (Encyclopedia B. Hosch, "machine learning", 2009. [En línea]. Disponible en: https://global.britannica.com/technology/machine-learning. [Consultado: 26-mar-2017].

[3] T. M. Mitchell, "The Discipline of Machine Learning", *Mach. Learn.*, vol. 17, núm. July, pp. 1–7, 2006.

[4] F. A. González, "Introducción Aprendizaje de Máquina", 2007. [En línea]. Disponible en: http://dis.unal.edu.co/profesores/fgonza/courses/2007-I/ml/ml-01-introduction.pdf. [Consultado: 11-jun-2017].

[5] Jason Brownlee, "Supervised and Unsupervised Machine Learning Algorithms", 2016. [En línea]. Disponible en: http://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/. [Consultado: 06-jun-2017].

[6] S. Abney, "Semisupervised Learning for Computational Linguistics", *J. R. Stat. Soc. Ser. A Stat. Soc.*, vol. 172, núm. 3, pp. 694–694, 2008.

[7] M. Vishal y S. Sabri, "Reinforcement Learning", *Machine Learning for Humans*, 2017. [En línea]. Disponible en: https://medium.com/machine-learning-for-humans/reinforcement-learning-6eacf258b265. [Consultado: 14-abr-2018].

[8] Jason Brownlee, "Discover Feature Engineering, How to Engineer Features and How to Get Good at It", 2014. [En línea]. Disponible en: http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/. [Consultado: 03-jun-2017].

[9] C. C. AGGARWAL y C. ZHAI, *Mining Text Data*, vol. 8. New York: Springer Science & Business Media, 2012.

[10] F. Kateb y J. Kalita, "Classifying Short Text in Social Media: Twitter as Case Study", *Int. J. Comput. Appl.*, vol. 111, núm. 9, pp. 1–12, 2015.

[11] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), "¿Qué es un blog?" [En línea]. Disponible en: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo_1_Iniciacionblog/qu_es_un_blog.html. [Consultado: 11-jun-2017].

[12] "LEY 1712 DE 2014", *D. Of. 49084 marzo 6 2014*, 2014.

[13] Julián Pérez Porto y Ana Gardey, "Definición de Twitter", 2014. [En línea]. Disponible en: http://definicion.de/twitter/. [Consultado: 11-jun-2017].

[14] G. Song, Y. Ye, X. Du, X. Huang, y S. Bie, "Short Text Classification: A Survey", *J. Multimed.*, vol. 9, núm. 5, pp. 635–643, 2014.

[15] Twitter International Company, “Política de Privacidad de Twitter”, 2017. [En línea]. Disponible en: https://twitter.com/privacy?lang=es#update. [Consultado: 11-jun-2017].

[16] T. C. Niño Sandoval, S. V. Guevara Pérez, F. A. González, R. A. Jaque, y C. Infante Contreras, “Uso de redes neuronales artificiales en predicción de morfología mandibular a través de variables craneomaxilares en una vista posteroanterior / Use of Artificial Neural Networks for Mandibular Morphology Prediction through Craniomaxillar Variables...”, *Univ. Odontol.*, vol. 35, núm. 74, 2016.

[17] I. Rish, “IBM Research Report An empirical study of the naive Bayes classifier”, 2001.

[18] Codecademy, “What is REST?”, 2018. [En línea]. Disponible en: https://www.codecademy.com/articles/what-is-rest. [Consultado: 07-abr-2018].

[19] K. Beck *et al.*, “Manifiesto por el Desarrollo Ágil de Software”. [En línea]. Disponible en: http://agilemanifesto.org/iso/es/manifesto.html. [Consultado: 06-ago-2017].

[20] P. Letelier y M. C. Penadés, “Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”.

B. ANEXO: SDD-1804-Componente de clasificación

En este anexo se realiza la especificación de elementos de diseño para el componente de software desarrollado en este proyecto

Componente de Clasificación de Textos Cortos Basado en Técnicas de Aprendizaje Automático

DOCUMENTO DE DISEÑO DE SOFTWARE

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Abril, 2018	Juan Jiménez	1.0	

Introducción

Propósito

Este documento busca brindar información correspondiente al diseño del componente de software *Beta Classification* a aquellos interesados en su uso o complementación de desarrollo; acá se detallarán los elementos necesarios para el funcionamiento del componente.

Arquitectura del Sistema

En esta sección, se darán a conocer las diferentes vistas de la arquitectura del componente de software.

Vista Lógica del Sistema

Diagrama de paquetes

En este diagrama se detallan los paquetes definidos para el funcionamiento del componente de software, el cual se basa en la estructura propuesta por el patrón MVC (Modelo, Vista, Controlador), sin embargo, se reemplaza el paquete modelo por un paquete de persistencia, el paquete de persistencia se encarga de la gestión de archivos que se ven involucrados en el programa, el controlador maneja la lógica de negocio, y en el paquete vista se descarga la responsabilidad de la presentación del sistema.

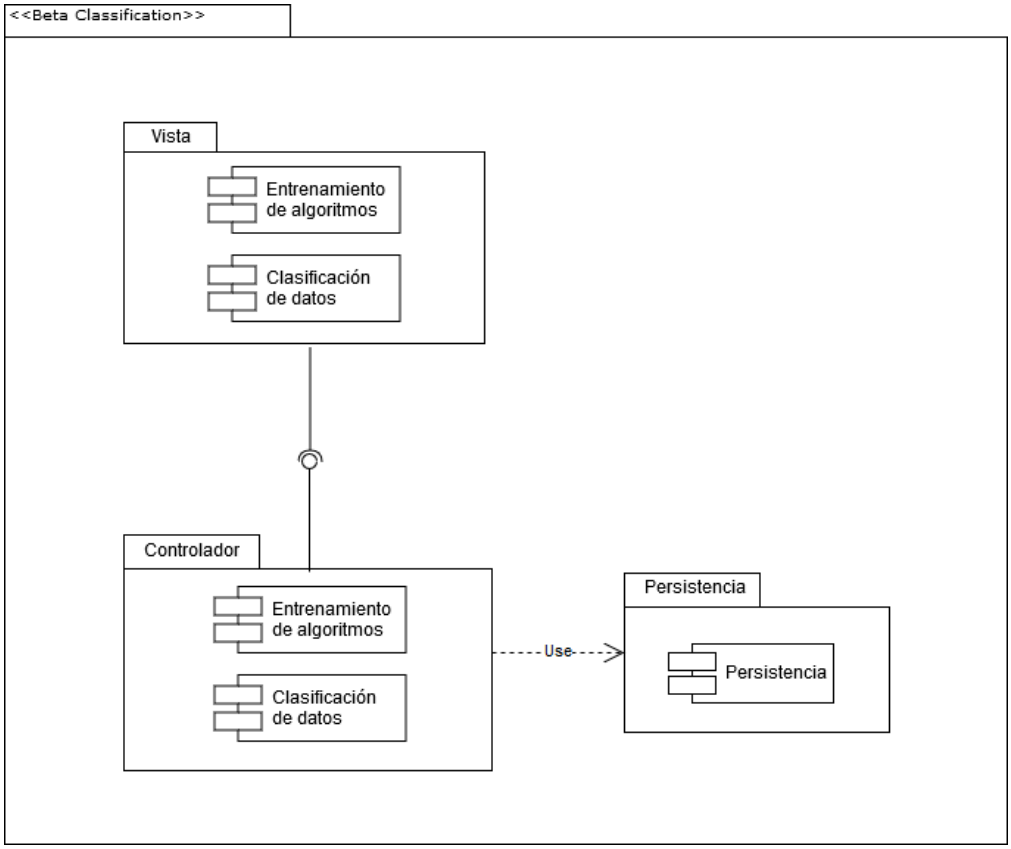


Ilustración 1 Diagrama de paquetes

Vista Física del Sistema

Diagrama de despliegue

En este diagrama se muestra la infraestructura del componente de software, que para este caso se basa en una arquitectura cliente servidor, se usará para la comunicación el protocolo HTTP, y las funcionalidades se exponen a través de micro servicios REST, permitiendo así su uso desde la interfaz propia de la herramienta provista en este desarrollo, o a través de llamados directos al API del componente.

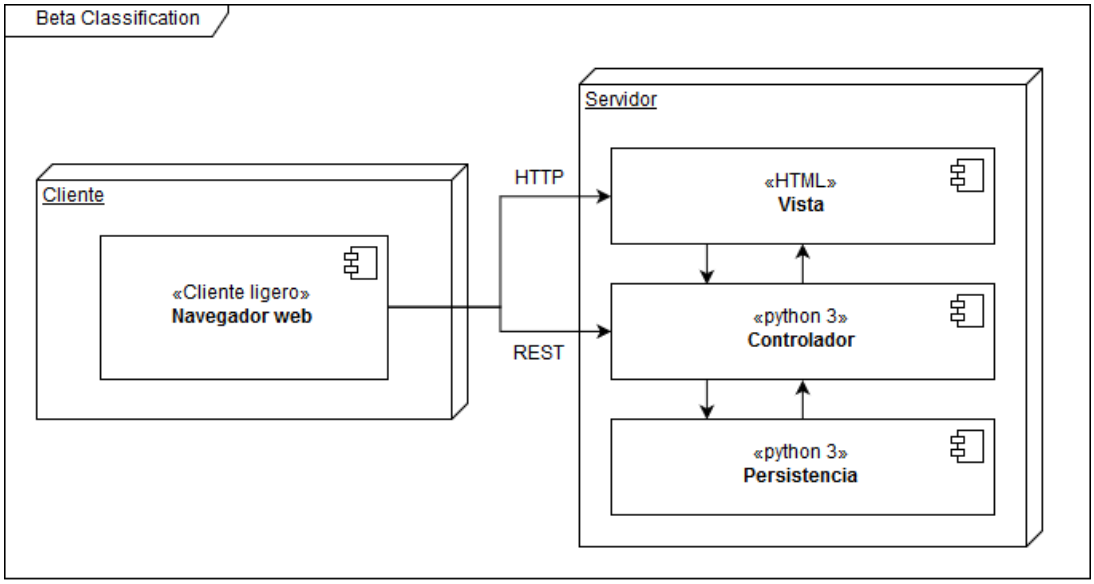


Ilustración 2 Diagrama de despliegue

Vista de Procesos del Sistema

Diagramas de proceso

En esta sección se detallan los procesos que realizará el componente *Beta Classification,* se describe por lo tanto el comportamiento de los procesos de entrenamiento de los algoritmos y clasificación de datos a continuación.

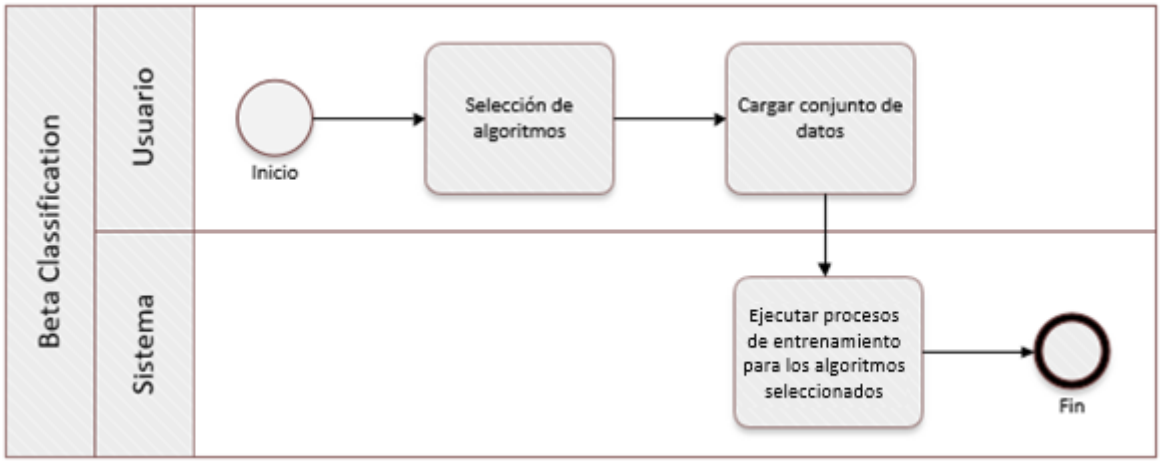


Ilustración 3 Proceso de entrenamiento de algoritmos

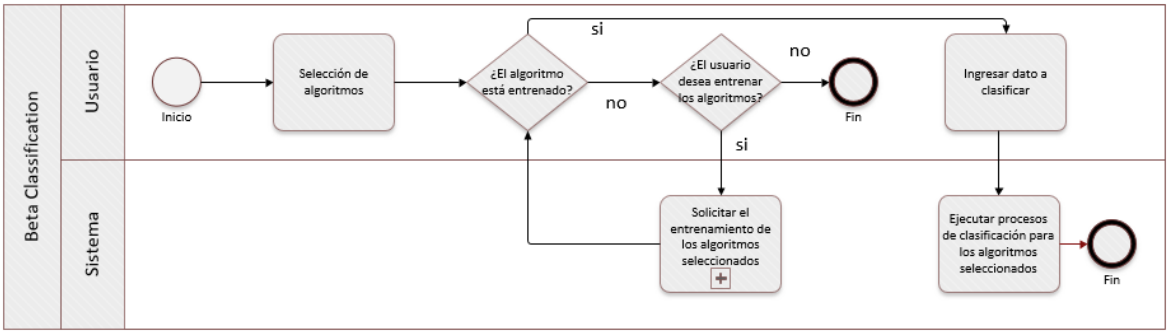


Ilustración 4 Proceso de clasificación de datos

Diseño detallado

Comportamiento del Sistema

Diagrama de secuencia

A continuación, se detallan las secuencias que se realizan en el componente de software al momento de realizar las operaciones solicitadas de entrenamiento de algoritmos y clasificación de datos.

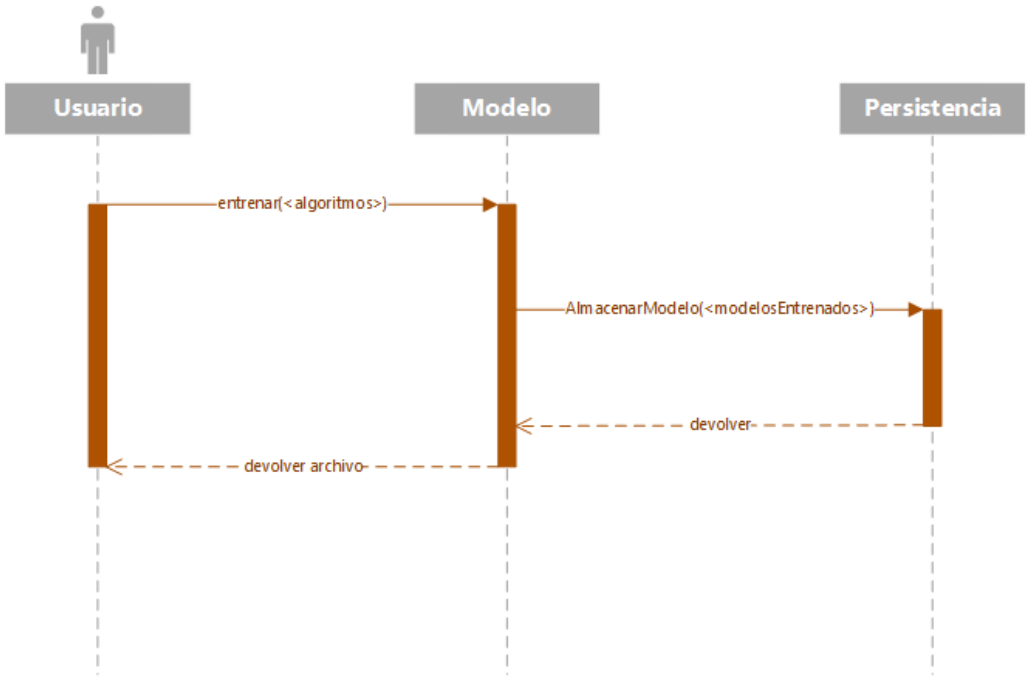


Ilustración 5 Diagrama de secuencia del proceso de entrenamiento

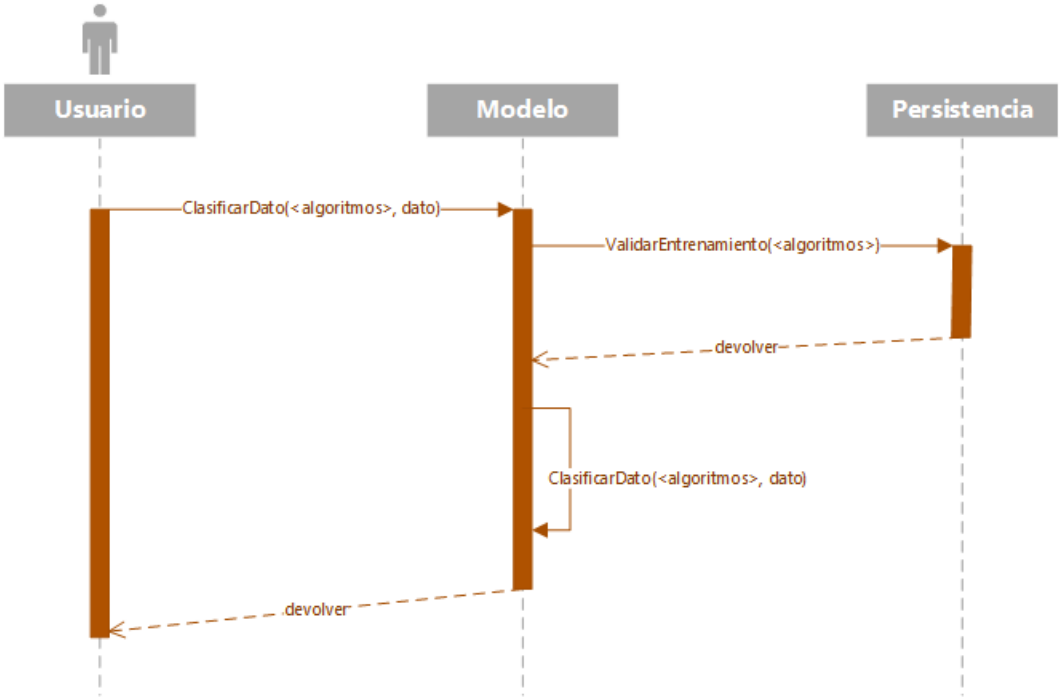


Ilustración 6 Diagrama de secuencia del proceso de clasificación

Interfaz de Usuario

Diagrama de navegabilidad

Se plantea el siguiente diagrama de navegabilidad para que los usuarios hagan uso de la herramienta.

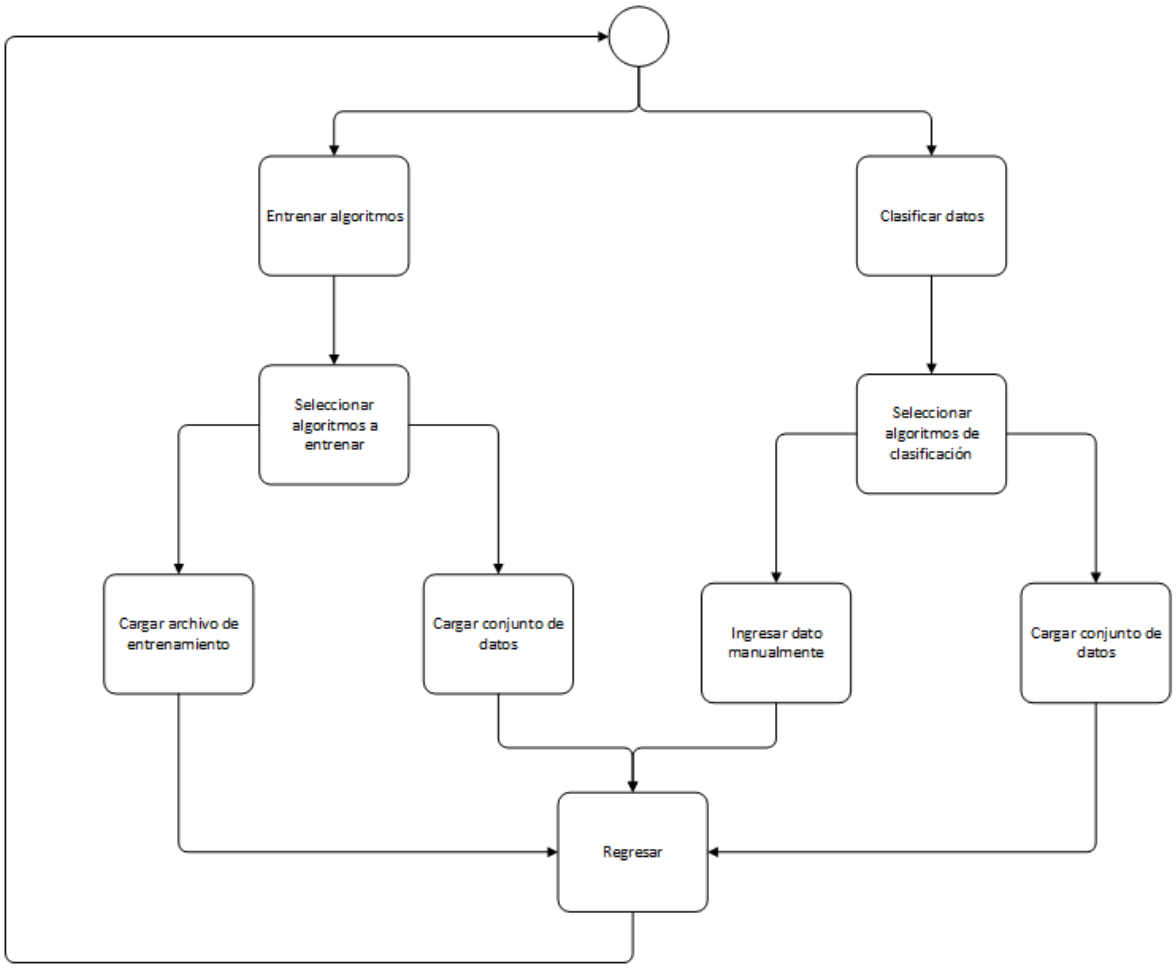


Ilustración 7 Diagrama de navegabilidad del componente de software

C. ANEXO: SAD-1804-Componente de clasificación

En este anexo se realiza la definición de la arquitectura para el componente de software desarrollado en este proyecto

Componente de Clasificación de Textos Cortos Basado en Técnicas de Aprendizaje Automático

DOCUMENTO DE ARQUITECTURA DE SOFTWARE

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Abril, 2018	Juan Jiménez	1.0	Definición inicial la arquitectura del software

Introducción

Propósito

El presente documento hablará de la arquitectura que hace parte del componente de analítica para clasificación de textos con el fin de que pueda ser integrable como parte del framework propuesto para el macro proyecto: “Efectividad de un protocolo de re experimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política”.

Alcance

El alcance se basa en describir la arquitectura del componente de analítica que se desarrollará teniendo en cuenta las diferentes capas que lo componen y así mismo las diferentes vistas que permitan modelar las funcionalidades que tendrá el software.

Arquitectura

La arquitectura de software es muchas veces vista como un puente entre los requerimientos del sistema y la implementación de los mismos. Como bien es sabido, para la realización de un software se cuentan con tres tipos de arquitectura: Monolítica, Cliente-Servidor y tres capas.

En este caso, la arquitectura que va a trabajarse para el desarrollo del componente de analítica es la arquitectura Cliente - Servidor o arquitectura RestFull, en donde el software puede repartirse la carga computacional que tiene en dos partes que a su vez son completamente independientes una de la otra. Por el lado del servidor es donde se proveen los recursos o servicios; y por el lado del cliente se tiene en consideración los usuarios participantes.

Gracias a la implementación de esta arquitectura, es posible asegurar un mayor control sobre el software teniendo en cuenta que pueda ser fácilmente escalable y que pueda realizarse un mantenimiento del mismo más fácilmente.

De acuerdo a la teoría anteriormente expuesta, el componente a realizar maneja esta arquitectura de la siguiente forma:

Representación de la arquitectura

Punto de vista de Stakeholder

El punto de vista de Stakeholder, explica las posibles relaciones entre los que se encuentran participando en las diferentes actividades que se ven relacionadas con el software. Esta vista cuenta con los Stakeholder, Valoraciones (importancia en los resultados) y objetivos que exponen la finalidad de cada uno de los Stakeholder.

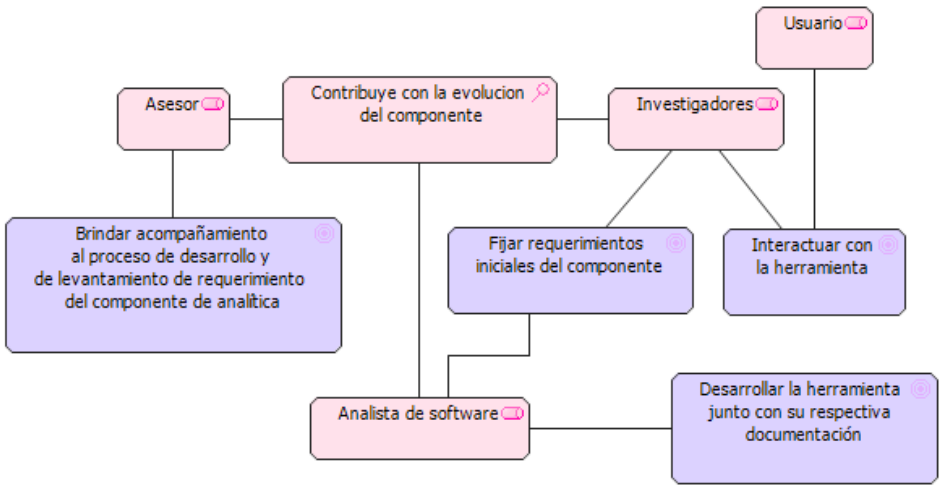


Ilustración 1: Punto de vista de Stakeholder

Para este caso particular los realmente interesados o involucrados en el desarrollo del software son los investigadores, el asesor, el analista de software y los usuarios finales, cada uno de ellos tiene un objetivo que les permitirá contribuir con el mismo fin.

Punto de vista de función negocio

En esta vista es posible ver cada una de las funciones que le corresponden a los diferentes participantes y sus relaciones en términos de flujo de información.

Enfocándose en el componente a desarrollar, los actores, que a su vez cuentan con su respectivo rol, cuentan con diferentes funciones que a su vez llevan a nuevas acciones, lo que hace que se pueda observar la relación que tienen ellos en cuanto a las funciones que tienen que desarrollar dentro de la realización de la herramienta.

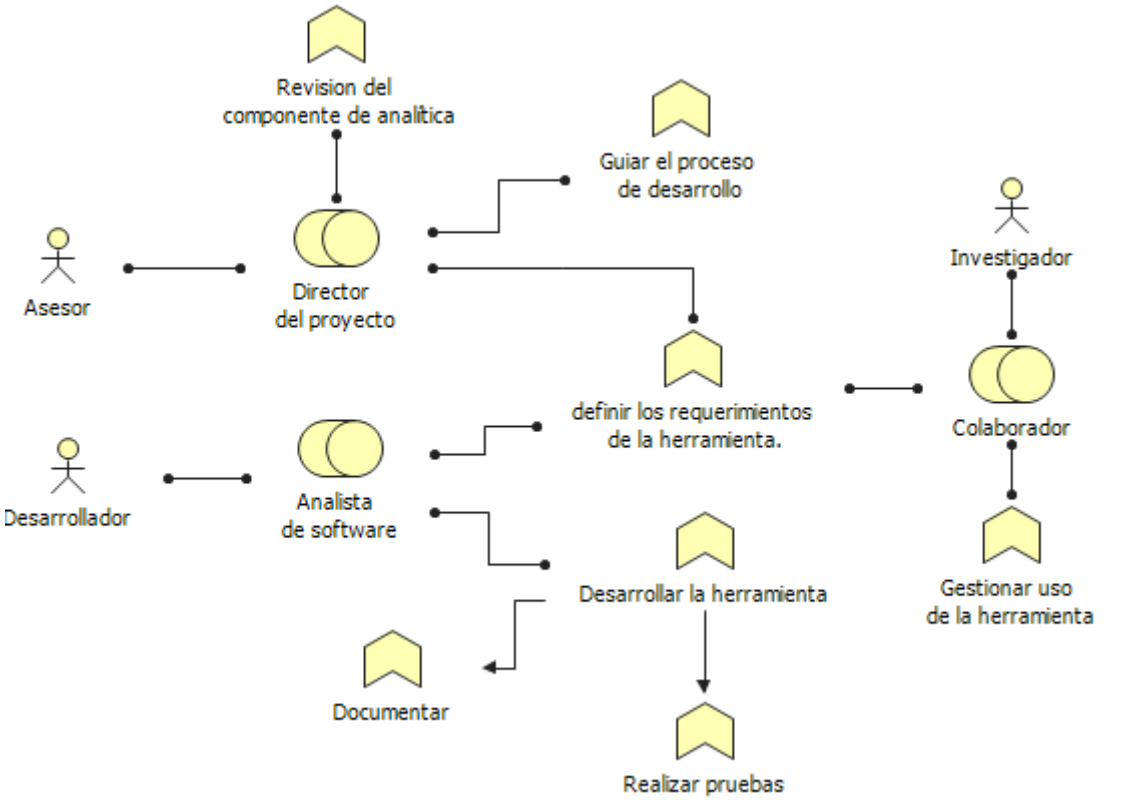


Ilustración 2: Punto de vista de función de negocio

Punto vista de cooperación de la aplicación

El punto de vista de cooperación de aplicación nos muestra el funcionamiento de la aplicación teniendo en cuenta la localización desde la cual se desempeña cada componente del software.

Se centra más que todo en la comunicación entre los componentes u otras interfaces y las funciones que cumple cada uno de ellos. En este caso se tienen cuatro módulos principales, tres de ellos pertenecen al back office y la parte de interfaz web pertenece al front end. En el diagrama es posible observar cada una de las comunicaciones existentes entre módulos, lo que permite entender un poco más cómo funciona la herramienta.

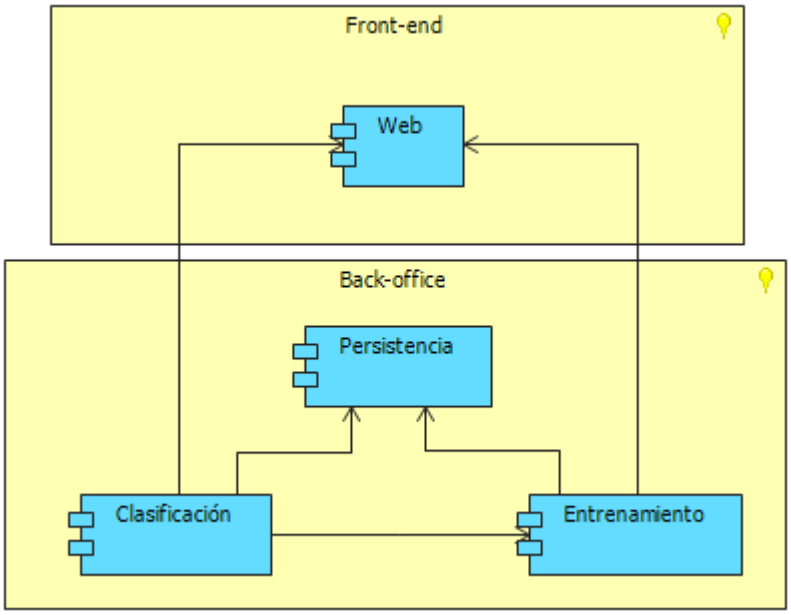


Ilustración 3: Punto de vista de cooperación de aplicación

Punto vista de uso de la aplicación

En esta vista se plantea un proceso como eje central el cual es capaz de desplegar diferentes funciones que ofrece en este caso la herramienta. De igual manera este punto de vista permite pre-visualizar apartados donde se encontraran los servicios de despliegue y estandarización dentro del desarrollo, lo que permite la asignación de funciones, responsabilidades, objetivos, medios y metodologías.

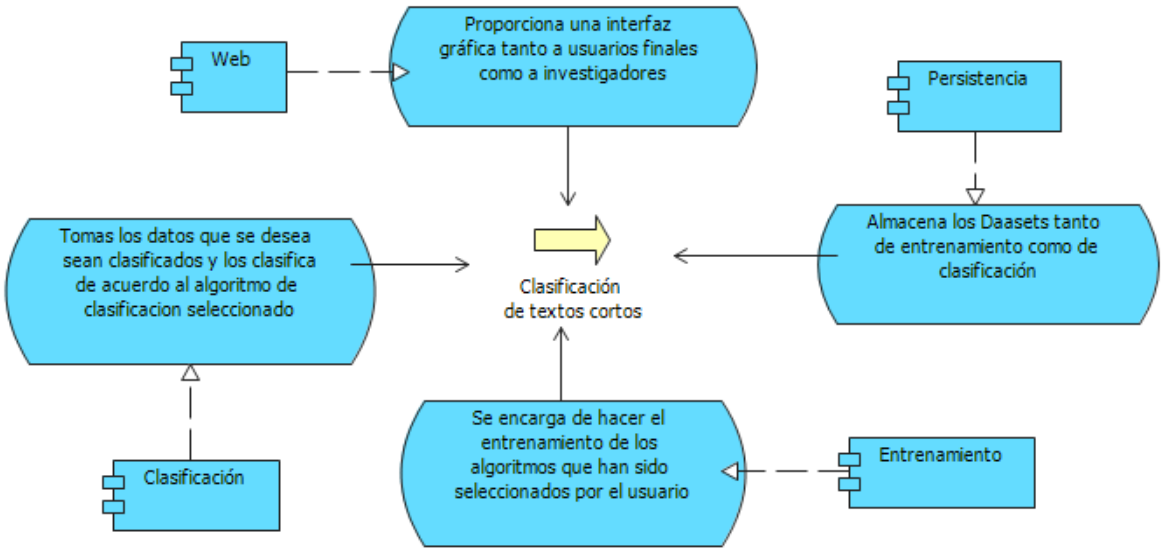


Ilustración 4: Punto de uso de la aplicación

De igual forma, se tienen los cuatro módulos base de la herramienta, cada uno de ellos presta servicios de acuerdo a la función que cada uno tiene. Todos estos servicios giran en torno a al proceso central de clasificación de textos cortos que es el eje central y punto final del desarrollo de la herramienta.

Punto vista de infraestructura

Aquí es posible visualizar la infraestructura que se debe tener para el desarrollo de la herramienta y así poder realizarse un servicio de infraestructura o derivar nuevas funciones de infraestructura y de esta forma tener un flujo entre las mismas. De igual forma, la función va unida con un nodo, que es el que se encuentra en la capacidad de unirse a una determinada locación, y el cual va a ser especializado por un dispositivo que se encuentre conectado a este y que tiene un acceso bidireccional con la der que realiza una ruta de comunicación nuevamente con dicho nodo. Éste, se compone de una interfaz de infraestructura y así mismo es usado por ella para poder establecer una unión entre dichos componentes y el servicio que se va a prestar.

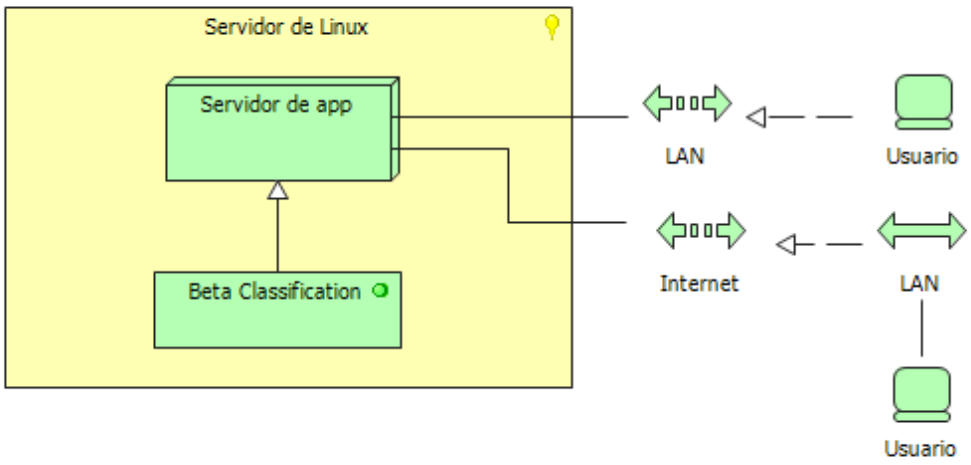


Ilustración 5: Punto de vista de infraestructura

En este caso se puede observar que se tienen dos tipos de usuario: un usuario que se conecta desde internet, y otro usuario que una red LAN que lo lleva directamente al servidor. Este usuario envía peticiones al servidor de aplicaciones el cual se conecta con la herramienta como tal. Todo esto se ve contenido en un servidor de Linux.

Punto vista de Organización e implementación

Esta vista muestra cómo se realizan una o más aplicaciones en la infraestructura teniendo en cuenta el mapeo que se realice de las aplicaciones, componentes lógicos, artefactos físicos y el mapeo de la información utilizada por dichas aplicaciones y componentes en la infraestructura de almacenamiento

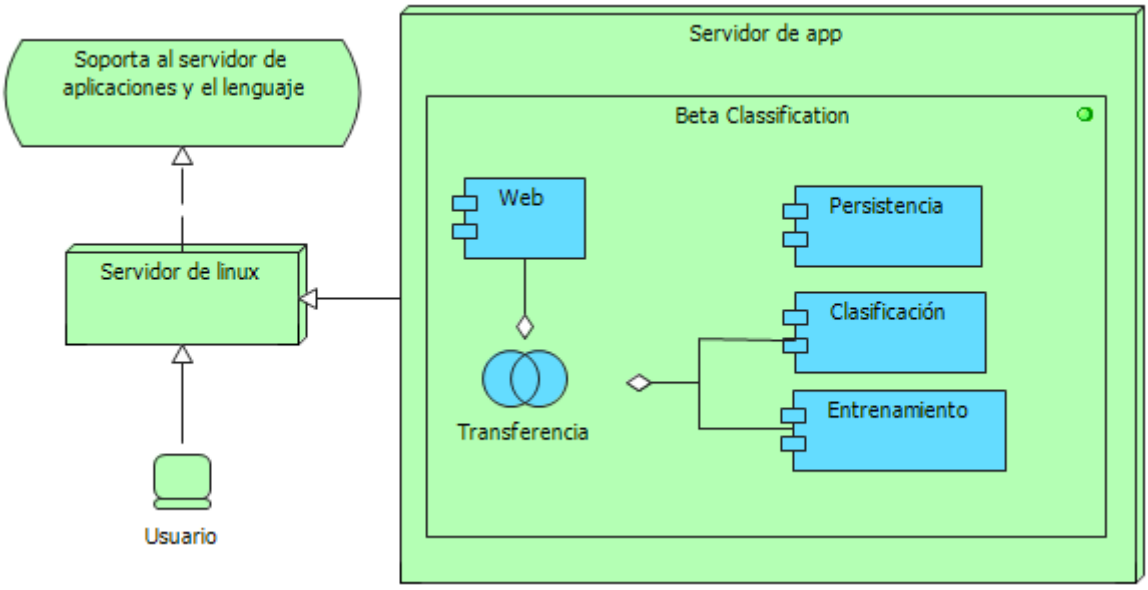


Ilustración 6: Punto de vista de organización e implementación

Esta vista es un complemento de la anteriormente presentada, ya que en ella se observan los módulos que hacen participes de la infraestructura de la herramienta como tal, teniendo en cuenta el flujo de información que se presenta entre los módulos por medio de una colaboración de transferencia de datos.

Punto vista de proyecto

El punto de vista de proyecto se enfoca principalmente en observar la forma en como el proyecto se relaciona con los diferentes componentes esenciales para su desarrollo. En primera medida tenemos como punto central, el paquete de trabajo, que es aquel que encierra el proyecto en sí que se ha realizado. Este, realiza una serie de liberables o entregables, los cuales, son la parte visible de los resultados del proyecto. Así mismo, el paquete de trabajo realiza un objetivo planteado al inicio de su desarrollo. En lo referente al actor y al rol que este desempeña en el proyecto, estos vienen unidos a el paquete de trabajo, el cual, tiene consigo mismo un fijo de trabajo.

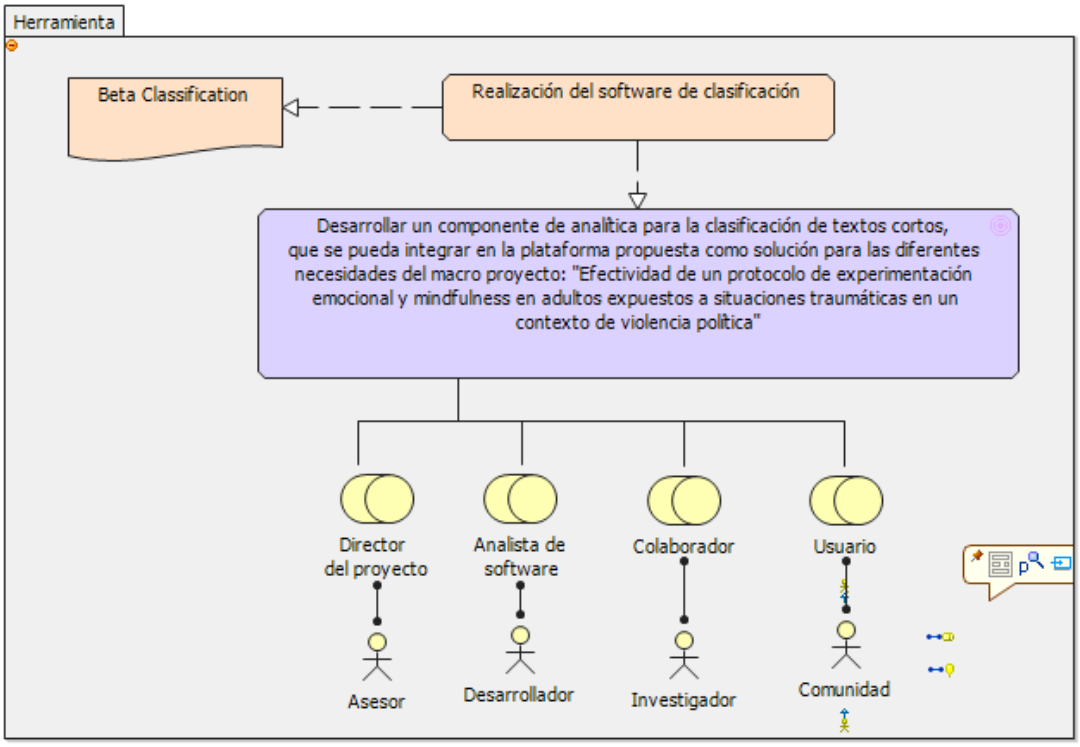


Ilustración 7: Punto de vista de proyecto

En este caso de paquete de trabajo se tiene la clasificación de textos cortos ya que es en este punto en el que se centra toda la herramienta, y es de este punto donde surge el liberable que es el componente “Beta Classification” teniendo en cuenta el objetivo inicial planteado y los actores que hacen parte del proyecto.

D. ANEXO: STD-1806-Componente de clasificación

En este anexo se realiza la definición y reporte de las pruebas de funcionamiento realizadas al componente de software

Componente de Clasificación de Textos Cortos Basado en Técnicas de Aprendizaje Automático

DOCUMENTO DE PRUEBAS

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Junio, 2018	Juan Jiménez	1.0	Descripción del documento y definición inicial de las pruebas realizadas funcionales aplicadas al componente.

Introducción

Este documento busca realizar una definición clara de las pruebas de aceptación realizadas sobre el componente de software mediante las cuales se comprueba el correcto funcionamiento de este componente.

Propósito

Identificar las pruebas funcionales que se requieren para asegurar la calidad del componente de software y el cumplimiento de los requerimientos definidos.

Alcance

Identificar las pruebas funcionales que se requieren para asegurar la calidad del componente de software y el cumplimiento de los requerimientos definidos, así como los resultados de la aplicación de las mismas, en este documento no se realizarán las pruebas de integración con otras de plataformas debido a falta de acceso a las mismas.

Definiciones, Acrónimos y Abreviaturas

Definiciones

Datasets: Conjunto de datos.

Acrónimos, Abreviaturas

ANN: Redes neuronales artificiales, Artificial Neural Network por sus siglas en inglés.
DT: Árbol de decisión, Decision Tree por sus siglas en inglés.
KNN: K más cercanos, K-Nearest Neighbors, por sus siglas en inglés.
NB: Clasificador ingenuo de Bayes, Naive Bayes por sus siglas en inglés.
SVM: Máquina de soporte vectorial, Support Vector Machine por sus siglas en inglés.

Proceso de pruebas

Casos de prueba

A continuación, se presentan los casos de prueba generales que se definieron para ser aplicados al componente de software:

Código	CP_001
Nombre del caso de prueba	Cargar de datos para entrenamiento.
Funcionalidad a probar	Carga de conjuntos de datos al componente.
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir la carga de un conjunto de datos, siempre y cuando se encuentre dentro de los formatos permitidos (txt, csv, xls y xlsx).
Criterios	**Aceptación:** El componente de software permite cargar satisfactoriamente el archivo que contiene el conjunto de datos. **Falla:** No se puede cargar el archivo de datos, o se permite cargar un archivo con una extensión invalida.
Precondiciones	Archivo que contiene el conjunto de datos.
Requisitos del caso de prueba	El conjunto de datos debe tener encabezado ya que para su carga se requiere el nombre de determinadas columnas.
Proceso	Desde la interfaz del componente de software, dar clic en el menú de entrenamiento, allí solicita la selección del archivo, y las columnas donde se contienen las clases y el texto. *(para poder continuar con la prueba se debe llenar el formulario completo)*.
Postcondiciones	Una vez seleccionado el fichero y ejecutado el proceso de carga, el mismo debe estar disponible para el uso en el componente.

Código	CP_002
Nombre del caso de prueba	Cargar de datos para clasificación.
Funcionalidad a probar	Carga de conjuntos de datos al componente.

Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir la carga de un conjunto de datos, siempre y cuando se encuentre dentro de los formatos permitidos (txt, csv, xls y xlsx).
Criterios	**Aceptación:** El componente de software permite cargar satisfactoriamente el archivo que contiene el conjunto de datos. **Falla:** No se puede cargar el archivo de datos, o se permite cargar un archivo con una extensión invalida.
Precondiciones	Archivo que contiene el conjunto de datos, algoritmos previamente entrenados o archivos de salida del proceso de entrenamiento.
Requisitos del caso de prueba	El conjunto de datos debe tener encabezado ya que para su carga se requiere el nombre de determinadas columnas.
Proceso	Desde la interfaz del componente de software, dar clic en el menú de clasificación, allí solicita la selección del archivo, y la columna donde se contiene el texto a clasificar. *(para poder continuar con la prueba se debe llenar el formulario completo)*.
Postcondiciones	Una vez seleccionado el fichero y ejecutado el proceso de carga, el mismo debe estar disponible para el uso en el componente.

Código	CP_003
Nombre del caso de prueba	Definir atributo de salida al entrenar.
Funcionalidad a probar	Definir atributo de salida.
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir indicar cual es la columna que contiene las clases durante la carga de un conjunto de datos para el entrenamiento.
Criterios	**Aceptación:** El componente de software permite especificar el nombre de la columna que contiene las diferentes clases. **Falla:** No se puede escribir el nombre de la columna que contiene las clases, o se permite continuar sin definir dicha columna.
Precondiciones	Archivo que contiene el conjunto de datos.

Requisitos del caso de prueba	El conjunto de datos debe tener encabezado ya que para su carga se requiere el nombre de determinadas columnas.
Proceso	Desde la interfaz del componente de software, escribir el nombre de la columna donde se contienen las clases. *(para poder continuar con la prueba se debe llenar el formulario completo)*.
Postcondiciones	El componente debe continuar con el proceso de entrenamiento.

Código	CP_004
Nombre del caso de prueba	Entrenar NB.
Funcionalidad a probar	Entrenar clasificador Naive Bayes (NB).
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir el entrenamiento del clasificador ingenuo de Bayes.
Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza el entrenamiento del clasificador ingenuo de Bayes y entrega como resultado un archivo en donde se contienen los parámetros de entrenamiento. **Falla:** No se permite entrenar el algoritmo, o no se entrega el archivo de resultado.
Precondiciones	Dataset de entrenamiento.
Requisitos del caso de prueba	Aprobar correctamente los casos de prueba CP_001 y CP_003.
Proceso	Desde la interfaz de entrenamiento de los algoritmos seleccionar el clasificador ingenuo de Bayes, y completar el formulario de entrenamiento.
Postcondiciones	El componente debe entregar los resultados del entrenamiento mediante una matriz de confusión y un archivo con la configuración del algoritmo entrenado.

Código	CP_005
Nombre del caso de prueba	Entrenar DT.
Funcionalidad a probar	Entrenar clasificador Árbol de Decisión (DT).

Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir el entrenamiento del clasificador árbol de decisión.
Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza el entrenamiento del clasificador seleccionado y entrega como resultado un archivo en donde se contienen los parámetros de entrenamiento. **Falla:** No se permite entrenar el algoritmo, o no se entrega el archivo de resultado.
Precondiciones	Dataset de entrenamiento.
Requisitos del caso de prueba	Aprobar correctamente los casos de prueba CP_001 y CP_003.
Proceso	Desde la interfaz de entrenamiento de los algoritmos seleccionar el clasificador árbol de decisión, y completar el formulario de entrenamiento.
Postcondiciones	El componente debe entregar los resultados del entrenamiento mediante una matriz de confusión y un archivo con la configuración del algoritmo entrenado.

Código	CP_006
Nombre del caso de prueba	Entrenar SVM.
Funcionalidad a probar	Entrenar clasificador Máquina de Soporte Vectorial (SVM).
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir el entrenamiento del clasificador máquinas de soporte vectorial.
Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza el entrenamiento del clasificador seleccionado y entrega como resultado un archivo en donde se contienen los parámetros de entrenamiento. **Falla:** No se permite entrenar el algoritmo, o no se entrega el archivo de resultado.
Precondiciones	Dataset de entrenamiento.
Requisitos del caso de prueba	Aprobar correctamente los casos de prueba CP_001 y CP_003.

Proceso	Desde la interfaz de entrenamiento de los algoritmos seleccionar el clasificador máquinas de soporte vectorial, y completar el formulario de entrenamiento.
Postcondiciones	El componente debe entregar los resultados del entrenamiento mediante una matriz de confusión y un archivo con la configuración del algoritmo entrenado.

Código	CP_007
Nombre del caso de prueba	Entrenar ANN.
Funcionalidad a probar	Entrenar clasificador Red Neuronal Artificial (ANN).
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir el entrenamiento del clasificador red neuronal artificial.
Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza el entrenamiento del clasificador seleccionado y entrega como resultado un archivo en donde se contienen los parámetros de entrenamiento. **Falla:** No se permite entrenar el algoritmo, o no se entrega el archivo de resultado.
Precondiciones	Dataset de entrenamiento.
Requisitos del caso de prueba	Aprobar correctamente los casos de prueba CP_001 y CP_003.
Proceso	Desde la interfaz de entrenamiento de los algoritmos seleccionar el clasificador red neuronal artificial, y completar el formulario de entrenamiento.
Postcondiciones	El componente debe entregar los resultados del entrenamiento mediante una matriz de confusión y un archivo con la configuración del algoritmo entrenado.

Código	CP_008
Nombre del caso de prueba	Entrenar KNN.
Funcionalidad a probar	Entrenar clasificador k-nearest neighbors (KNN).
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe permitir el entrenamiento del clasificador KNN.

Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza el entrenamiento del clasificador seleccionado y entrega como resultado un archivo en donde se contienen los parámetros de entrenamiento. **Falla:** No se permite entrenar el algoritmo, o no se entrega el archivo de resultado.
Precondiciones	Dataset de entrenamiento.
Requisitos del caso de prueba	Aprobar correctamente los casos de prueba CP_001 y CP_003.
Proceso	Desde la interfaz de entrenamiento de los algoritmos seleccionar el clasificador KNN, y completar el formulario de entrenamiento.
Postcondiciones	El componente debe entregar los resultados del entrenamiento mediante una matriz de confusión y un archivo con la configuración del algoritmo entrenado.

Código	CP_009
Nombre del caso de prueba	Evaluar rendimiento.
Funcionalidad a probar	Generar gráficas de rendimiento.
Descripción	Según lo especificado en los requerimientos funcionales del componente, se debe comparar el rendimiento de los diferentes algoritmos de clasificación.
Criterios	**Aceptación:** El componente de software mediante el cumplimiento de las precondiciones realiza la clasificación de un dataset permitiendo comparar el tiempo invertido por cada algoritmo seleccionado en la tarea de clasificación. **Falla:** No se permite clasificar el dataset, o no se comparan los tiempos invertidos.
Precondiciones	Dataset con datos a clasificar, algoritmos entrenados.
Requisitos del caso de prueba	Aprobar correctamente alguno de los casos de prueba entre el CP_004 y CP_008.
Proceso	Desde la interfaz de clasificación se deben seleccionar los algoritmos que se desea realicen el proceso de clasificación **(Los cuales deben estar previamente entrenados)**, se

	completa el formulario de entrenamiento, y se da clic en el botón clasificar.
Postcondiciones	El componente debe entregar los resultados de la tarea de clasificación mediante archivos con la clase asignada para cada registro, y una gráfica en donde se evidencie el rendimiento de los algoritmos seleccionados.

Código	CP_010
Nombre del caso de prueba	Evaluar tiempo de respuesta.
Funcionalidad a probar	Tiempo de respuesta.
Descripción	Según lo especificado en los requerimientos no funcionales del componente, se debe clasificar un registro en tiempo que no exceda los 5 segundos.
Criterios	**Aceptación:** El componente de software toma como máximo 5 segundos en realizar la clasificación de un registro sin clasificar. **Falla:** El tiempo invertido por el clasificador es superior al máximo permitido.
Precondiciones	Dataset con datos a clasificar, algoritmos entrenados.
Requisitos del caso de prueba	Aprobar correctamente el caso de prueba CP_009.
Proceso	Al dar clic en el botón clasificar (CP_009), verificar los tiempos que toma a cada algoritmo realizar la clasificación del dataset y promediarlo en la cantidad de registros.
Postcondiciones	Ninguna.

Resultados de las pruebas

Código escenario de prueba.	Submódulo evaluado.	¿Aprobó?
CP_001	Entrenamiento	SI
CP_002	Clasificación	SI
CP_003	Entrenamiento	SI
CP_004	Entrenamiento	SI
CP_005	Entrenamiento	SI
CP_006	Entrenamiento	SI
CP_007	Entrenamiento	SI
CP_008	Entrenamiento	SI
CP_009	Clasificación	SI
CP_010	Clasificación	SI

E. ANEXO: Manual de instalación

En este manual se explica el paso a paso para la instalación y puesta en marcha de componente de software

Manual de instalación para el componente de software Beta Classification

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Abril, 2018	Juan Jiménez	1.0	Versión inicial del documento.
Mayo, 2018	Juan Jiménez	2.0	Se añaden capturas de pantalla sobre la ejecución.

Introducción

Propósito

Identificar los requerimientos para la instalación y puesta en marcha del componente de software *Beta Classification*.

Audiencia

Este documento en primera instancia se dirige a las personas que deseen poner en marcha de manera el componente de software, así como a aquellas que desee, mantener y/o actualizar el componente.

Autores

Nombre	Juan Sebastián Jiménez Valero
Rol	Analista de Software
Categoría Profesional	Estudiante de ingeniería de sistemas
Contacto	jsjimenez50@ucatolica.edu.co

Requerimientos de configuración

Servidor

El componente fue desarrollado en un servidor con sistema operativo CentOS7, utilizando el lenguaje de programación Python 3 y el framework Flask.

Preparación del servidor

Con el fin de poder realizar las operaciones de instalación requeridas en el servidor se deberá contar con una cuenta con permisos de súper usuario, se recomienda que no sea la cuenta root.

Realizaremos la instalación de las herramientas de desarrollo y el cliente git de CentOS para poder compilar código fuente mediante el comando.

```
sudo yum –y groupinstall development
sudo yum –y git
```

El flag –y sirve para que no se solicite confirmación de la instalación que se realizará, al finalizar esta ejecución debemos instalar Python 3 desde el repositorio IUM (*Inline with Upstream Stable*), para esto ejecutaremos los siguientes comandos en este orden:

```
sudo yum –y install https://centos7.iuscommunity.org/ius-release.rpm
sudo yum –y install python36u
```

Para finalizar instalaremos los paquetes pip y herramientas de desarrollo:

```
sudo yum –y install python36u-pip
sudo yum –y install python36u-devel
```

Creación del ambiente de desarrollo

Con el fin de aislar las configuraciones y archivos que se realicen específicamente para el proyecto se deben crear entornos ambientes de desarrollo.

```
mkdir code
cd code
python3.6 –m venv beta_classification
```

Este comando creará una carpeta llamada beta_classification, la cual almacenará todos los archivos relacionados con el proyecto, posterior a esto activamos el ambiente de desarrollo mediante la secuencia de comandos:

```
cd beta_classification
source bin/activate
```

Como resultado de estos comandos tendremos desde la consola lo siguiente:

```
(beta_classification) [jjimenez@juno beta_classification]$
```

Ilustración 1 Ambiente de desarrollo activado

Esto indica que se activó el entorno beta_classification de manera correcta, cuando dejemos de trabajar en el hacemos uso del comando `deactivate`.

Entorno de desarrollo

Como entorno de desarrollo para el proyecto se eligió el trabajo con Visual Studio Code, el cual está disponible en la página `https://code.visualstudio.com/download`. Elegimos la descarga para el SO indicado, para el caso de CentOS, aplicará la descarga del archivo rpm correspondiente a RedHat, como se ve en la ilustración 3, al descargar el archivo damos clic en la opción guardar, como se ve en la ilustración 4.

Ilustración 2 Archivo a descargar Visual Studio Code

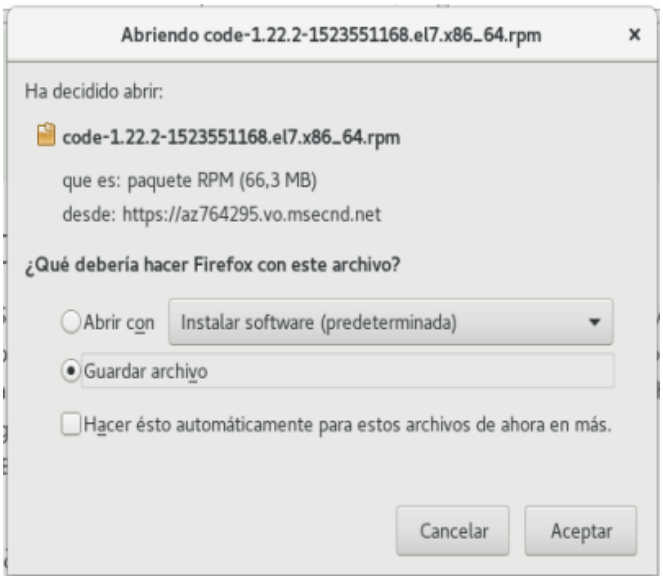


Ilustración 3 Opciones de descarga

Una vez finalizada la descarga ejecutamos el paquete con el comando:

```
sudo rpm –ivh code-1.22.2-1523551168.el7.x86_64.rpm
```

Una vez instalado Visual Studio Code, instalamos la extensión para trabajar con Python de la siguiente manera, desde la consola de comandos de Visual Studio Code escribimos.

```
ext install ms-python.python
```

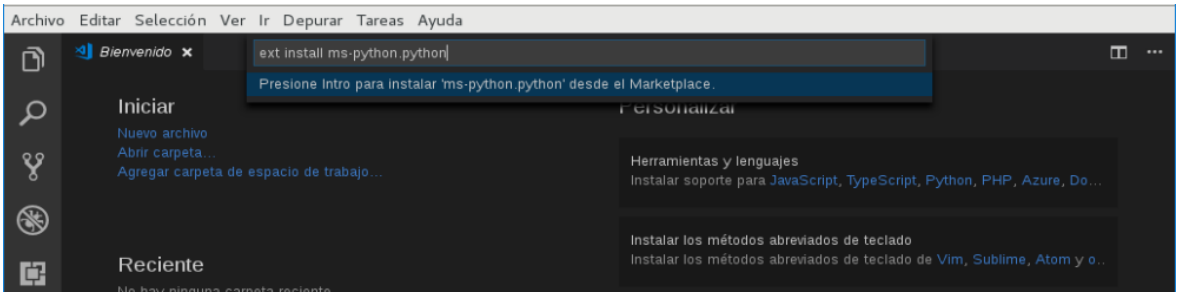


Ilustración 4 Instalación del complemento de Python para Visual Studio Code

En el enlace `https://marketplace.visualstudio.com/items?itemName=ms-python.python` encontrará más información sobre el complemento.

Extensiones y librerías requeridas

Flask

Para la instalación del framework Flask debemos activar primero el ambiente de desarrollo que creamos, a continuación, ejecutaremos el comando:

```
sudo pip3.6 install uwsgi flask
```

Flask-WTF

Complemento usado para la validación de formularios, y su instalación se realiza así:

```
sudo pip3.6 install flask-WTF
```

Flask-Uploads

Este complemento será usado para la carga de datasets, y su instalación se realiza así:

```
sudo pip3.6 install flask-uploads
```

Pandas

Esta librería será usada para análisis de datos, y su instalación se realiza así:

```
sudo pip3.6 install pandas
```

Matplotlib

Esta librería será usada para la generación de gráficas, y su instalación se realiza así:

```
sudo pip3.6 install matplotlib
```

Scikit-Learn

Mediante esta librería se soportarán los componentes de aprendizaje del componente, su instalación requiere que se instale también la librería scipy, por lo tanto, se deben ejecutar los siguientes comandos:

```
sudo pip3.6 install scipy
sudo pip3.6 install scikit-learn
```

NLTK

Esta librería es importante para la etapa de pre procesamiento, su comando de instalación es:

```
sudo pip3.6 install nltk
```

una vez se haya instalado debemos ejecutar la siguiente secuencia de comandos para descargar el componente requerido.

```
python3.6
import nltk
nltk.download('all')
```

Imbalanced-Learn

Mediante esta librería se realiza el balanceo de las clases contenidas en el *data set*, el comando de instalación es:

```
sudo pip3.6 install -U imbalanced-learn
```

Clonación del código fuente

Como herramienta de desarrollo colaborativo, se creó un repositorio en GitHub, en el cual se almacenará todo el código fuente del componente, para ello se utilizará la instalación realizada de la herramienta Git, de la siguiente forma:

```
git clone https://github.com/JJimenez94/beta_classification.git
```

Como resultado, este comando descargará todo el código fuente disponible en el repositorio.

Ejecución del componente

Una vez acá se debe abrir la carpeta que contiene el código fuente y damos clic derecho, ejecutar en terminal.

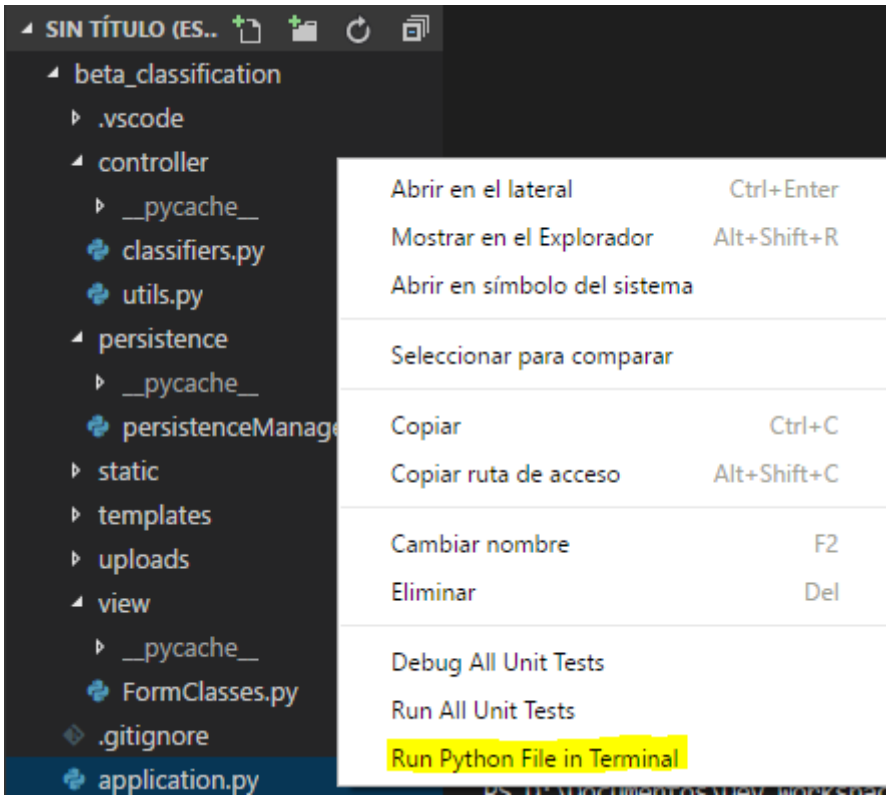


Ilustración 5 Ejecución del proyecto

Como resultado tendremos la aplicación lista para ser accedida a través de un navegador web.

```
* Serving Flask app "application" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Ilustración 6 Confirmación de la ejecución

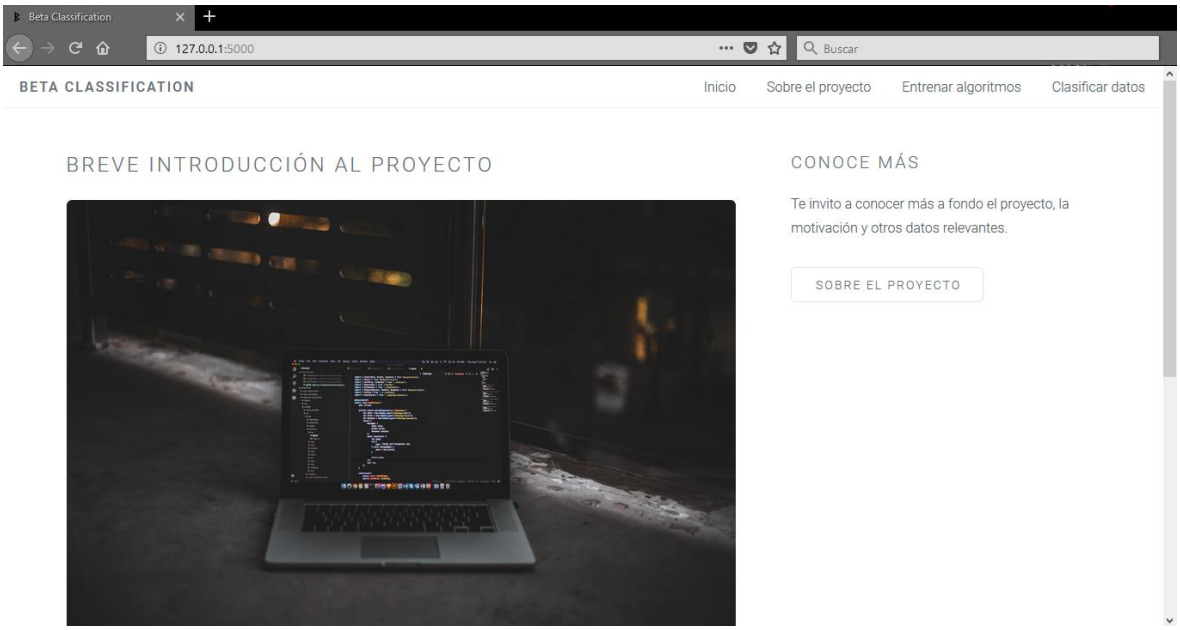


Ilustración 7 Pantalla inicial del módulo

F. ANEXO: Manual de usuario

En este manual se brinda una guía sobre la interacción que pueda tener el usuario con el componente de software.

Manual de usuario para el componente de software Beta Classification

JUAN SEBASTIÁN JIMÉNEZ VALERO

Contenido

Control de Cambios

Fecha	Autor	Versión	Comentarios
Mayo, 2018	Juan Jiménez	1.0	Versión inicial del documento.

Introducción

Beta Classification es un componente de software que se encarga de realizar la clasificación de textos cortos, mediante el uso de algoritmos de aprendizaje supervisado, el componente fue pensado para cumplir con las tareas de clasificación de textos cortos originadas en el proyecto de investigación institucional "Efectividad de un protocolo de reexperimentación emocional y mindfulness en adultos expuestos a situaciones traumáticas en un contexto de violencia política", y durante el planteamiento del mismo, se definió que debía clasificar textos cortos aunque se encontraran fuera del contexto de este proyecto de investigación, siempre y cuando se realice un debido proceso de clasificación.

Propósito

Este documento pretende dar a los usuarios que van a interactuar con el componente de software una guía sobre su funcionamiento y las posibles restricciones que tuvieran lugar en dicha interacción.

Audiencia

Este documento en primera instancia se dirige a las personas que deseen interactuar con el componente de software.

Autores

Nombre	Juan Sebastián Jiménez Valero
Rol	Analista de Software
Categoría Profesional	Estudiante de ingeniería de sistemas
Contacto	jsjimenez50@ucatolica.edu.co

Desarrollo del documento

Información general

La interfaz gráfica del componente de software es una interfaz web, la cual debe ser accedida a través de un navegador, se realizaron pruebas accediendo al módulo a través de los navegadores Mozilla Firefox, Google Chrome y Microsoft Edge. Las capturas de pantalla presentadas en el presente documento se tomaron de la ejecución del componente en el navegador Mozilla Firefox.

Estructura base de las páginas

Todas las páginas web de esta interfaz cuentan con un esquema en común, el cual se compone de:

- Encabezado: Se presenta el nombre del proyecto junto con un menú con el cual se puede navegar en las opciones ofrecidas por el componente.

- Pie de página: En esta sección se presenta información general del componente, como el nombre del desarrollador, una declaración de derechos reservados, ya que al estar licenciado bajo una licencia de tipo Open Source tiene algunos derechos reservados, y un icono de la plataforma GitHub que contiene un enlace al código fuente del componente.

Página de inicio

En esta primera pantalla se brinda una breve introducción al componente.

Ilustración 1 Página de inicio

Sobre el proyecto

El objetivo de esta página es dar a conocer detalles técnicos de la elaboración del mismo, y la motivación para la realización del componente, se enuncian también los algoritmos que fueron implementados.

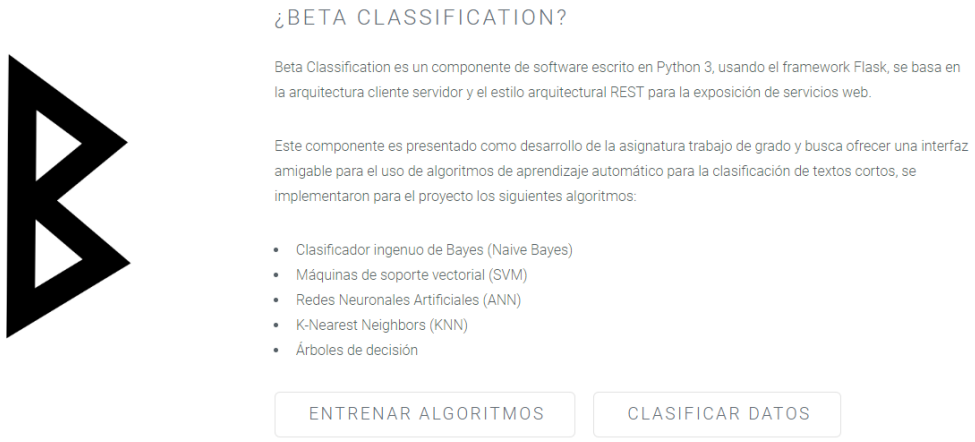


Ilustración 2 Página "sobre el proyecto"

Entrenar algoritmos

En esta página se brindan opciones para la carga de los modelos entrenados o en su defecto para iniciar un nuevo proceso de entrenamiento.

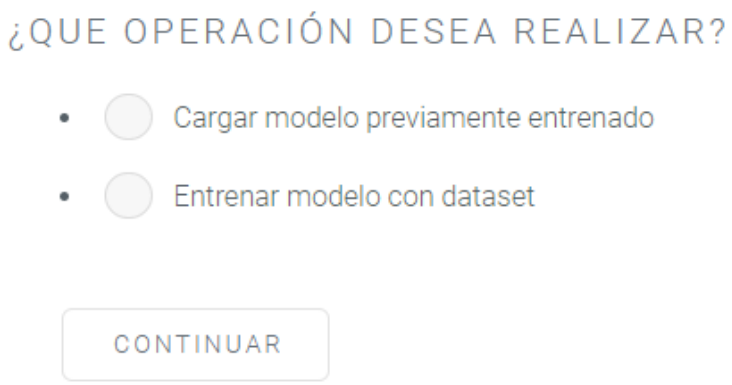


Ilustración 3 Página "entrenar algoritmos"

Cargar modelo previamente entrenado

Al seleccionar la opción de cargar modelo, se presentará un listado con los algoritmos implementados, de los cuales se deberá escoger uno, y a continuación dar clic en el botón seleccionar archivo para elegir el archivo que contiene el modelo entrenado, una vez se hayan cumplido con ambos requerimientos se debe dar clic en el botón continuar.

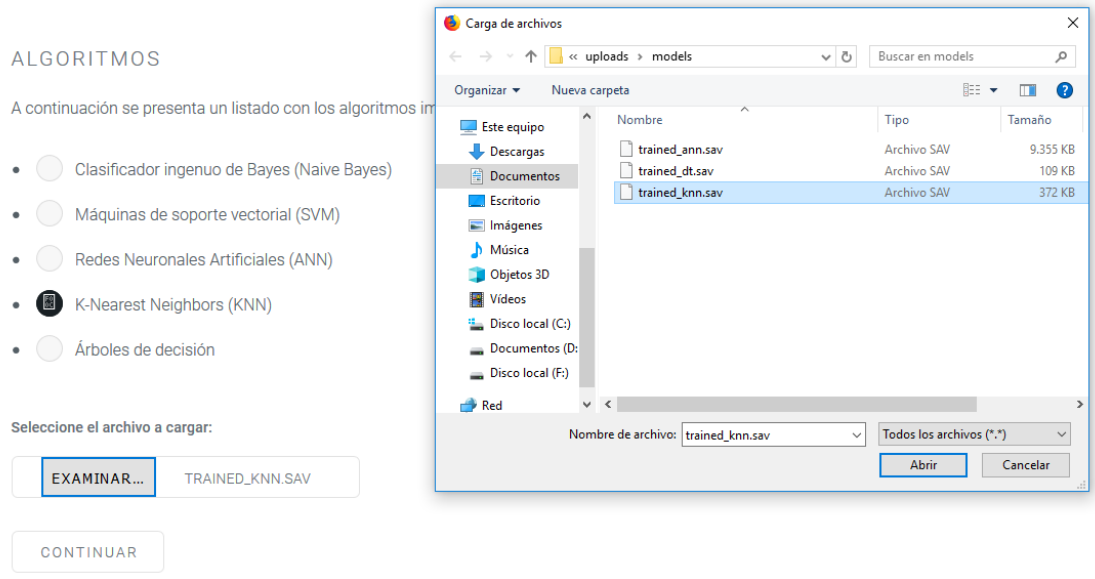


Ilustración 4 Carga de un modelo entrenado

Nota: solamente se deben cargar archivos con extensión *.sav que hayan sido generados por la aplicación.

Al finalizar la carga, la herramienta devolverá un mensaje confirmando la operación.

Ilustración 5 Resultados carga de un modelo entrenado

Entrenar modelo con dataset

Al seleccionar la opción de entrenar a partir de un dataset, se presentará el listado de los algoritmos implementados, de los cuales se deberá escoger al menos uno, esta opción permite entrenar más de un modelo a la vez. Para hacerlo debemos seleccionar el dataset con el que vamos a entrenar los algoritmos, y escribir los nombres de las columnas que contienen los datos y las clases en las casillas correspondientes.

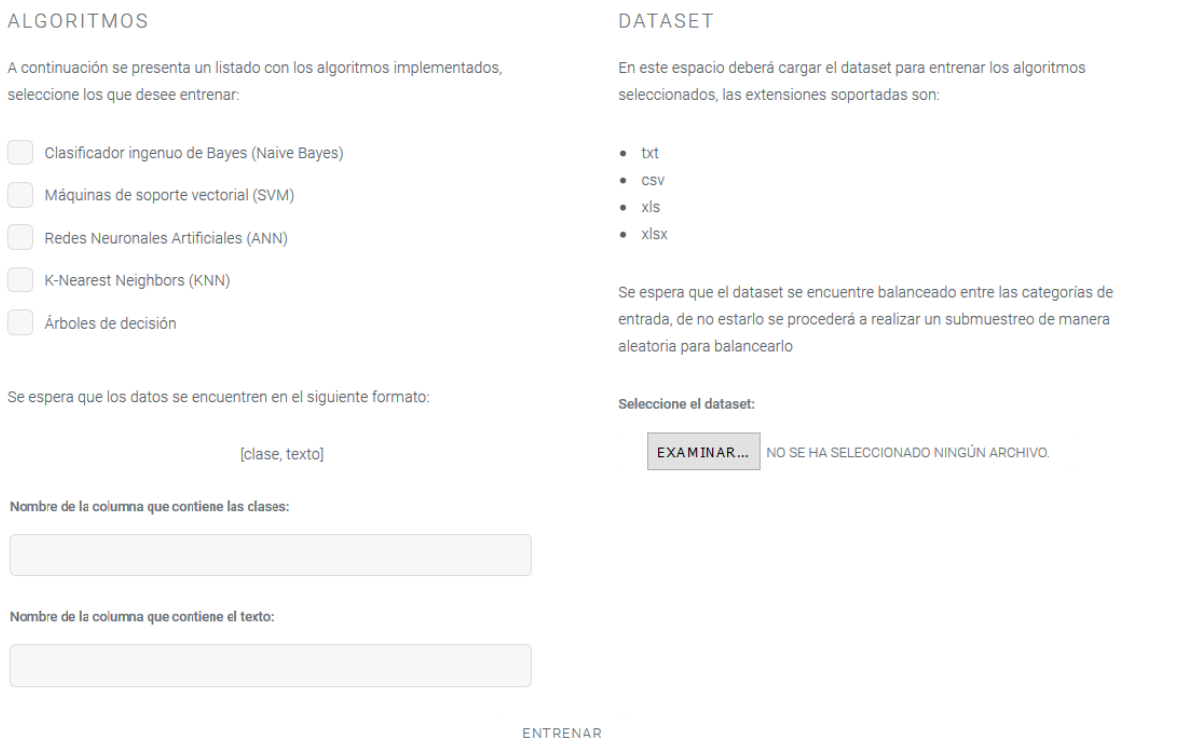


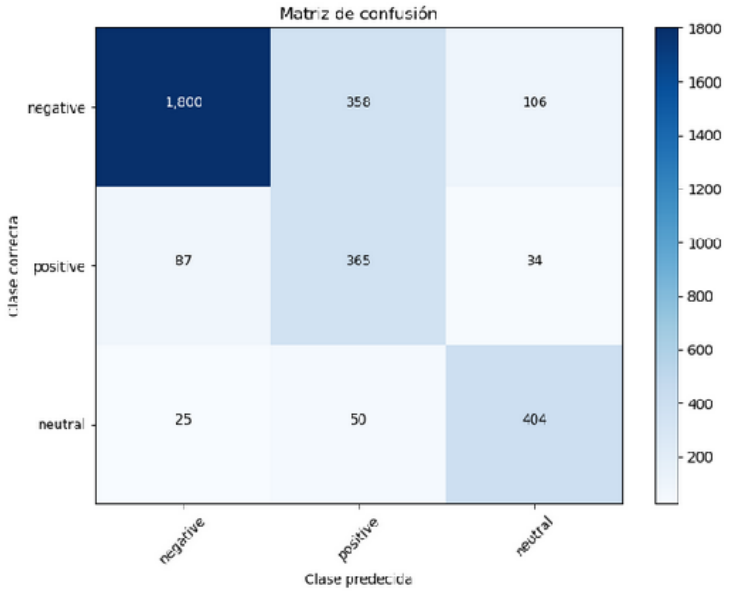
Ilustración 6 Entrenamiento a partir de dataset

Al finalizar la carga, la herramienta devolverá una página en donde se mostrará el resumen de los resultados obtenidos por este proceso de entrenamiento, y la matriz de confusión correspondiente para cada uno de los algoritmos seleccionados, cada resultado se mostrará así:

RESULTADOS DE ENTRENAR

REPORTE DE RESULTADOS PARA EL CLASIFICADOR MÁQUINAS DE SOPORTE VECTORIAL

Clase	precision	recall	f1	Cantidad de datos
negative	0.94	0.80	0.86	2264
neutral	0.47	0.75	0.58	486
positive	0.74	0.84	0.79	479
promedio/total	0.84	0.80	0.81	3229



Para descargar el modelo clasificado haga click. En este enlace

Ilustración 7 Resultado de entrenamiento de algoritmo SVM

Al dar clic en el enlace que se encuentra debajo de la matriz de confusión, se podrá guardar el modelo entrenado en cualquier ubicación.

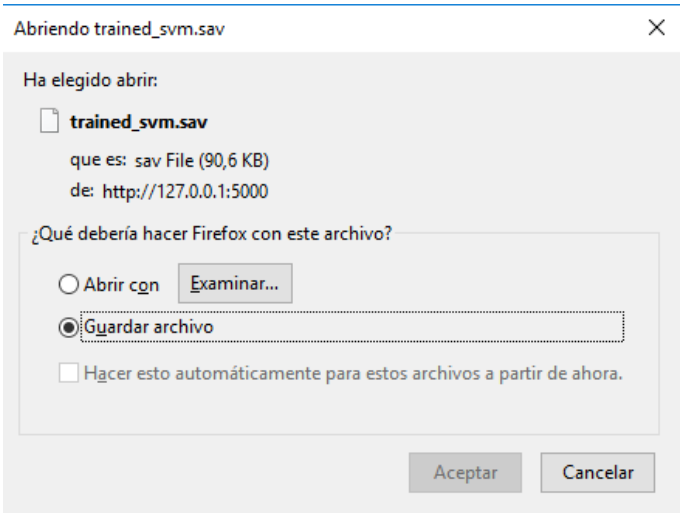


Ilustración 8 Almacenar modelo entrenado

Clasificar datos

En esta página se listan los algoritmos implementados, y se presenta un espacio para cargar el dataset que se desea clasificar, en este caso, se deberá ingresar el nombre de la columna que contiene los datos que se desean clasificar.

Ilustración 9 Página "Clasificar datos"

De igual forma que en la opción de entrenar a partir de un dataset podemos elegir varios algoritmos a la vez, cuando se haya concluido el proceso de clasificación se entregará un gráfico comparando el tiempo que le tomó clasificar los datos a cada algoritmo seleccionado, y el dataset resultante de las operaciones de clasificación.

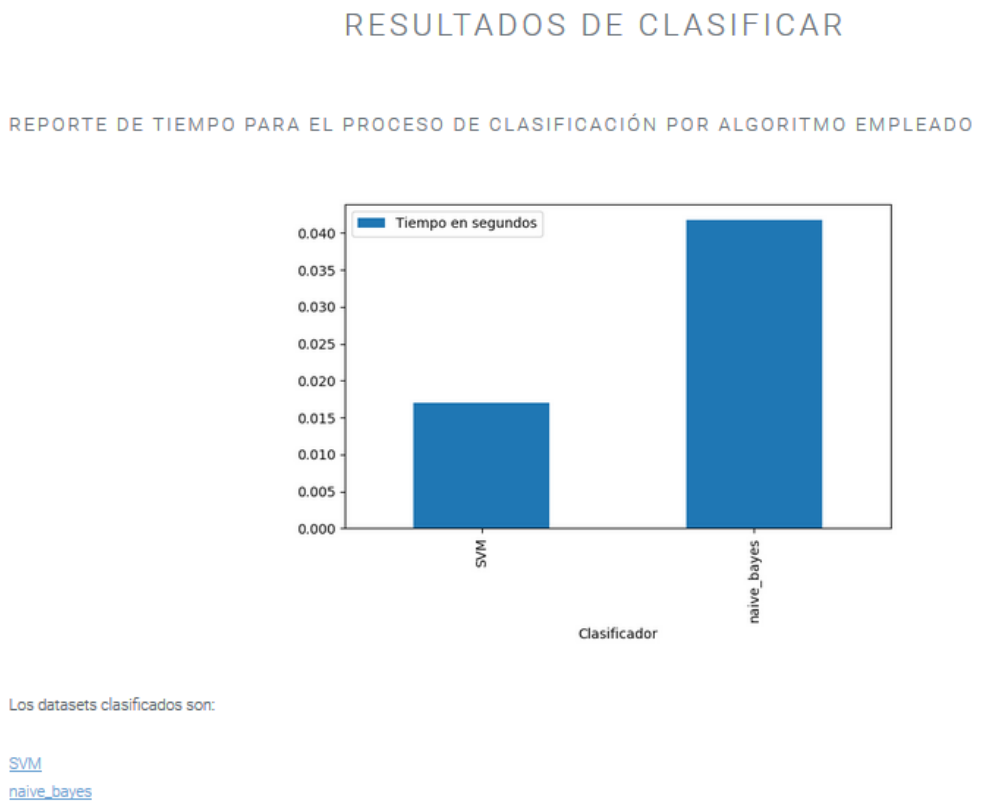


Ilustración 10 Resultados de clasificar datos