Universidad Autónoma Metropolitana

División de Ciencias de la Comunicación y Diseño
Departamento de Tecnologías de la Información

Licenciatura en Tecnologías y Sistemas de Información

Herramienta para visualizar similitudes de la personalidad entre usuarios en Twitter

Jorge Elías Vigil Santos

Asesor:

Mtra. Adriana Gabriela Ramírez de la Rosa

Co-asesor:

Dr. Esaú Villatoro Tello

Julio 2019

Índice

1. Introducción

La personalidad se define como las características de comportamientos, patrones de pensamientos y emocionales que se desarrollan a partir de factores biológicos y del entorno [1]. Para describir la personalidad formalmente, existen teorías basadas en rasgos, que definen la personalidad como una serie de características específicas del individuo; así como teorías basadas en comportamiento. En este proyecto se hace uso de una descripción basada en rasgos, conocida como el modelo Big Five. Este modelo, como se analizará posteriormente, descompone la personalidad en cinco factores, o rasgos. Estos rasgos son Extroversión, Apertura a nuevas experiencias, Estabilidad Emocional, Amabilidad y Responsabilidad, como se muestran en la figura 1.

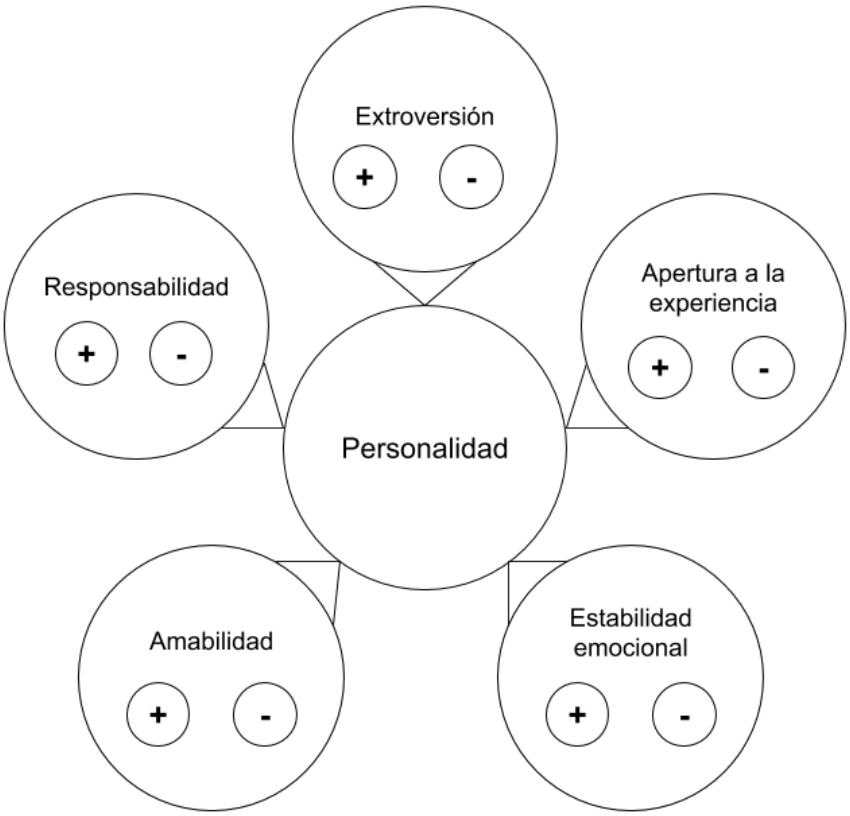

Figura 1: Rasgos de personalidad según el modelo Big-Five. Cada rasgo puede tener una polaridad positiva (+) o negativa (-).

Uno de los aspectos más importantes de la investigación psicológica es que permite al individuo una mejor comprensión sobre sí mismo. Se ha demostra-

do que el comportarse de manera opuesta a la disposición natural puede causar fatiga cognitiva, inhabilidad para usar energía para otras tareas además de mantener un comportamiento contrario al natural [28].

Otra ventaja del estudio de la personalidad es que ha permitido relacionar rasgos de ésta con dificultades para conducirse de manera emocionalmente estable. Por ejemplo, se ha identificado en los niños una relación entre rasgos como la introversión y la tendencia a desarrollar depresión en la vida adulta [10]. De manera similar, se encontró que altos niveles de sociabilidad y bajos niveles de timidez en los niños (que serían equiparables al rasgo de extroversión en adultos), se traducen en la tendencia a experimentar emociones positivas y satisfacción con la vida [10]. Por su parte, son particularmente interesantes los estudios que relacionan los rasgos de la personalidad con aspectos de salud [24].

El conocer aspectos de la personalidad puede ser beneficioso para los individuos que desean conducirse mejor en aspectos sociales. Sobre todo aquellos que presentan bajos niveles de extroversión, ya que su aversión al conflicto y tendencia a la ansiedad social los vuelve propensos a ciertos padecimientos como la depresión [14].

Cabe destacar el papel de la personalidad al entablar relaciones de amistad, un estudio relaciona los rasgos de la personalidad con el nivel de satisfacción con las relaciones de amistad que reportan los participantes [25]. Se reporta que mientras altos niveles de extroversión, responsabilidad y amabilidad se asocian a satisfacción con las relaciones de amistad y un bajo nivel de estabilidad emocional se asocia con menor satisfacción con las amistades. Por su parte, la apertura no mostró relación con la satisfacción en amistades.

Otros estudios asocian características específicas respecto a la calidad de las relaciones de amistad. Por ejemplo, un estudió descubrió que la apertura a la experiencia está asociada a una mayor probabilidad de tener amistades de diferente género y etnia, es decir, a tener un grupo de amigos más diverso. Este estudio también asocia un alto nivel de amabilidad y extroversión con "vínculos de amistad más tradicionales" [12]. Existen otras características encontradas de este estudio sobre la cantidad de amigos de acuerdo al perfil de personalidad. Por ejemplo, personas con baja extroversión, responsabilidad y apertura tienden a tener un solo amigo. Mientras aquellos con alta extroversión tienden a tener más amigos.

El estudio de las similitudes en las relaciones es un campo interesante de estudio. Las amistades tienden a ser similares en una amplia variedad de características, tales como edad, nivel de educación, raza, género y opiniones [27]. Resulta importante entonces, ligar la relación entre la personalidad de los individuos con el nivel de afinidad, o similitud, en sus relaciones de amistad.

Un estudio reciente ha descubierto que las personas tienen una similitud

de personalidad considerable en relación a sus amigos [27]. Esta similitud se ve reflejada en el uso de redes sociales. Los patrones de comportamiento que presentan los individuos en estas redes tienen una relación con los rasgos de su personalidad. Al asociar el comportamiento con los rasgos de personalidad, se encontró que amigos y personas en relaciones románticas cuentan con rasgos de personalidad similares.

En otro trabajo [5], se encontró que, al buscar por amistades, las personas tienden a elegir a aquellos con características similares. Este estudio tiene implicaciones en la manera de entender las relaciones interpersonales, sobre todo cuando las personalidades son diferentes. Se demostró que los desacuerdos, diferencias de personalidad, valores y preferencias pueden quebrantar la armonía de una relación. Así también, este estudio muestra que, en un ambiente donde la selección de amigos es limitada (por ejemplo, en una escuela pequeña), las personas muestran una tendencia a desarrollar menos relaciones de amistad. Esto debido a que resulta difícil encontrar amistades con características similares cuando hay pocas personas que comparten el espacio.

Los estudios mencionados anteriormente han beneficiado a varios sectores, desde el ámbito laboral hasta los campos de la salud. Debido a esto, se han explorado diversas posibilidades para la extracción de un perfil de personalidad. Es decir, obtener una descripción formal de la personalidad de un individuo.

El perfilado de personalidad ha sido de gran interés para la comunidad de investigación psicológica. Muchos campos se han beneficiado con el trabajo en este tema. En las ciencias forenses, por ejemplo, se pueden identificar y predecir tendencias criminales [8]. Por su parte, en el ámbito clínico, se ha podido asistir en el diagnóstico y la identificación de enfermedades mentales como la depresión [15].

Históricamente, la obtención del perfil psicológico se ha realizado de varias maneras. En un principio, se requería que los entrevistados redactaran ensayos auto-descriptivos. Varios especialistas evaluaban los textos para emitir juicios sobre la personalidad del autor de cada ensayo. Este proceso era lento, poco confiable y costoso [17]. Históricamente, la técnica de evaluación que más éxito ha tenido, y por ende la más usada, es la aplicación de pruebas estandarizadas. Los cuestionarios más usados en estas pruebas son: el NEO-Personality-Inventory Revised (NEO-PI R), de 240 preguntas y el Big-Five Inventory (BFI) de 44 preguntas [7]. En estas pruebas, el evaluado debe responder, en una escala definida según el instrumento (p.e. del uno al cinco), qué tan identificado se siente con cada enunciado del cuestionario. Las pruebas basadas en cuestionarios suelen tener una ventaja con respecto a las autodescriptivas, debido a que las respuestas pueden evaluarse de manera cuantitativa, al realizar alguna ponderación de los valores de las respuestas. Por otro lado, en un ensayo autodescriptivo, un especialista debe examinar el texto y determinar características a partir de éste. El proceso de análisis cualitativo puede ser mucho más lento y costoso que apli-

car cálculos a los valores numéricos de las respuestas en un cuestionario.

Ahora bien, dentro del campo del Procesamiento de Lenguaje Natural, existe un área de investigación conocida como el *perfilado de autor*: la clasificación e identificación automática de características de una persona a través del análisis de un texto de su autoría. Ciertas tareas del perfilado se han apoyado del estudio psicológico del lenguaje para generar herramientas que faciliten el proceso de perfilado [17]. Estos estudios han logrado relacionar el uso del lenguaje con múltiples rasgos de personalidad.

El Intenet, sobre todo las redes sociales, han servido como medio para obtener información relevante para la investigación psicológica. Los estudios en este campo se benefician de poder realizar observaciones en tiempo real sobre estas redes. También ven reducidos sus costos de implementación y se ha incrementado la cantidad de análisis que se pueden llevar a cabo. Esta información disponible en redes sociales ha sido usada para varias investigaciones, como las que buscan relacionar el perfil de personalidad de los usuarios con sus interacciones en redes sociales [6].

A partir del análisis realizado a la literatura, el cual muestra la necesidad de desarrollar sistemas eficientes, se propone en el presente trabajo la generación de una herramienta que permita al usuario la identificación y visualización de su perfil de personalidad, de acuerdo al modelo de Big-Five y, con base en éste, realizar una comparación entre su perfil y el de otros usuarios, para tener entendimiento de los círculos sociales del usuario y su relación con estos. Lo anterior por medio de técnicas de Procesamiento de Lenguaje Natual y de Aprendizaje Automático.

1.1. Justificación

1.1.1. Apoyo a la investigación psicológica.

Los beneficios del estudio de la personalidad en el campo de la salud son considerables. Existen, por ejemplo, estudios que ligan la personalidad con la longevidad [16].

Existen estudios sobre el impacto de la personalidad tanto a nivel físico, como psicológico. Sobre el impacto físico, una vertiente interesante asocia ciertos rasgos de la personalidad con el desarrollo de enfermedades a largo plazo [24]. Los resultados de estos trabajos muestran que ciertos rasgos de personalidad, están asociados a una mejor salud y ausencia de enfermedad. A manera de ejemplo, una alta responsabilidad reduce considerablemente el riesgo de ataques al corazón, diabetes y artritis. Resultados similares se encuentran asociados a alta apertura. Por su parte el neuroticismo se asocia con un incremento de afecciones cardiacas, pulmonares y de presión alta [24].

En la vertiente sobre el impacto psiquiátrico de la personalidad, se han encontrado fuertes relaciones genéticas entre ciertos rasgos de la personalidad y trastornos psicológicos. Por ejemplo, se ha descubierto una fuerte relación entre la extroversión y el desorden de déficit de atención hiperactivo (ADHD por sus siglas en Inglés), así como entre la apertura y la esquizofrenia y desorden bipolar. Este mismo trabajo relaciona el neuroticismo con psicopatía internalizada (trastornos de depresión o ansiedad) [14].

El trabajo de investigación sobre rasgos de personalidad como predictores de salud es relativamente nuevo. Esta vertiente proporciona una oportunidad para aprovechar los instrumentos de medición de personalidad y apoyar a la predicción y diagnóstico de enfermedades, ya sea físicas o psicológicas.

Así también, investigación sobre la afinidad entre amigos es importante. Si bien las personas muestran una tendencia a buscar amigos con características similares, existe también la posibilidad de encontrar amistad en personas con personalidad opuesta. En el estudio [5], se propone que relacionarse con individuos con personalidad diferente puede ser beneficioso. Puesto que al tener amistades con diferente personalidad, expone al individuo a nuevas ideas, valores y perspectivas.

Resulta interesante relacionar la afinidad entre amistades con la personalidad. Es planteamiento de este proyecto, explorar la relación entre los círculos sociales de los usuarios en Twitter y sus respectivos rasgos de personalidad.

1.1.2. Desventajas históricas de la aplicación de pruebas estandarizadas.

El uso de las pruebas estandarizadas se ha extendido hasta convertirse en un punto de referencia para la industria. Pero su aplicación puede ser poco práctica por diversos factores. Uno de estos factores tiene que ver con la necesidad de ser evaluadas por uno o varios especialistas, a partir de cuyos veredictos se modela un perfil de personalidad [7].

Una forma en la que se realizan estas pruebas es por medio de escritos de auto reporte. El análisis de estos escritos muchas veces requiere el desarrollo de esquemas que reflejen el contenido del ensayo. Como etiquetas que describan detalles de la redacción, como errores de ortografía, correcciones realizadas por el autor, entre otras [20]. El desarrollo de estos esquemas de clasificación y representación puede ser complicado y consumir mucho tiempo y recursos para la parte evaluadora.

La aplicación de estas pruebas también podría resultar en frustración para los entrevistados, dado el largo tiempo que puede llevar [1]. Otro inconveniente

[1] https://online.concordia.edu/business-news/pros-and-cons-of-personality-tests/

en cuanto a las pruebas de auto reporte, es que los resultados podrían no reflejar la realidad. Un participante podría no proporcionar respuestas reales, con tal de adaptarse a aquello para lo que cree que está siendo evaluado[2], o bien, podría interpretar la pregunta incorrectamente debido a la redacción de ésta. Como vemos, el proceso de aplicación de pruebas resulta en un consumo excesivo de tiempo y de recursos para todas las partes interesadas. Con esto en mente, la implementación de herramientas automáticas de evaluación de personalidad que requieran una mínima interacción por parte del usuario y los evaluadores puede ofrecer apoyo para abordar estos problemas.

1.2. Objetivos

El objetivo general del presente trabajo es:

Determinar niveles de afinidad entre miembros de una comunidad en Twitter a través de la identificación de su personalidad.

Los objetivos específicos son:

1. Implementar un clasificador de textos para identificar la personalidad de un autor basado en el modelo Big Five.
2. Definir métricas de evaluación que permitan determinar similitudes.
3. Representar los niveles de afinidad encontrados entre usuarios, usando técnicas de visualización de datos.

1.3. Estructura del documento

El resto del documento está compuesto como sigue. En el marco teórico se presentan conceptos y la teoría necesaria para comprender el contexto del presente trabajo. Se abordan temas de aprendizaje automático, el modelo de personalidad Big-Five, perfilado de autor, técnicas de representación de textos y métricas de evaluación consideradas para este proyecto. En el apartado de trabajo relacionado, se describen algunos proyectos y herramientas existentes relacionados con el perfilado de personalidad. Así pues, se hace una comparación entre las características de éstas y las aportaciones de este proyecto.

Una sección importante de este documento describe el método propuesto. Donde se presentan las actividades para llevar a cabo el trabajo, que consta a su vez de dos etapas: etapa de experimentación y desarrollo de aplicación. En la primera, se describe la experimentación llevada a cabo para generar un modelo de predicción de rasgos de personalidad. Mientras que en la segunda se explica el procedimiento de desarrollo de la aplicación, sus características, despliegue y

[2]https://www.seattletimes.com/seattle-news/health/faking-your-type-to-pass-a-personality-test/

limitaciones. Por último, se tiene un capítulo dedicado a conclusiones sobre el trabajo.

2. Marco teórico

2.1. La personalidad

Uno de los objetos de estudio de este trabajo, es el modelado y predicción de la personalidad a partir de texto escrito. De acuerdo con psicólogos, la personalidad se define como el sistema de procesamiento que describe las respuestas de comportamiento persistentes del ser humano ante una amplia variedad de estímulos externos [2]. Ésta caracteriza a un individuo y está involucrada en los procesos de comunicación. También influye en la manera en la que el individuo interactúa con los demás.

Existen diversas propuestas que pretenden describir la personalidad. Los más aceptados se basan en rasgos específicos que definen cómo se llevan a cabo los procesos de pensamiento, sentimiento y acción [26].

El más utilizado de los modelos basados en características o rasgos es el *Big Five Model* o *Five-Factor Model*, introducido por Norman en 1963 [22]. Este modelo propone que la personalidad se compone de cinco características binarias, esto es, cada una con polaridad positiva o negativa. Estas características, son: Extroversión (Extrovesion), Estabilidad emocional (Emotional stability), Amabilidad/Agradabilidad (Agreeableness), Responsabilidad (Conscientiousness) y Aapertura a la experiencia (Openness). Las descripciones de cada una de estas características se presentan a continuación:

- *Extroversión* asociada a la energía, manifestación de emociones positivas, asertividad, y capacidad de socializar. El polo negativo de este rasgo se conoce como *introversión*.
- *Estabilidad emocional* está ligada al control de los impulsos y es mejor descrita con su polo opuesto *Neuroticismo (Neuroticism)*, que es la tendencia a experimentar emociones negativas tales como enojo, ansiedad o depresión.
- *Amabilidad* se refiere a la predisposición a ser amable, compasivo y cooperativo. Su opuesto se refiere a expresar poca o nula confianza en los demás y apatía.
- *Responsabilidad* es la tendencia a demostrar auto-disciplina, responsabilidad y enfocarse en los logros. Su opuesto es mostrar comportamiento espontáneo.
- *Apertura a la experiencia* esta ligada a la tendencia a apreciar ideas inusuales, expresar curiosidad, creatividad y mostrar preferencia por el cambio. Su opuesto es demostrar poca originalidad y ser poco flexible al cambio.

Como se mencionó anteriormente, el método más usado para evaluar por los factores que determinan la personalidad es la aplicación de pruebas estandarizadas. La más utilizada para identificar la polaridad de los rasgos de Big-Five

es el Big-Five inventory (BFI) [7].

Ahora, existe una prueba estandarizada conocida como TIPI (Ten-Item Personlaity Inventory), que surge de la necesidad de aplicar instrumentos de evaluación de personalidad de manera rápida [21]. Esta prueba consiste en diez preguntas para determinar los valores de los cinco rasgos del Big-Five. Cada elemento de TIPI consiste en un enunciado que describe características de personalidad. A manera de ejemplo, una pregunta de TIPI que evalua por el rasgo de extraversión sería: "Me veo a mi mismo como: extrovertido, entusiasta". Cada elemento se valora entonces en una escala del 1 (*muy en desacuerdo* al 7 (*muy de acuerdo*). Debido que TIPI consiste solo de diez preguntas, toma alrededor de un minuto en completarse. TIPI puede reportar los rasgos de personalidad de manera congruente con otras pruebas de reporte externo y de auto reporte; además de mostrar congruencia entre aplicaciones repetidas.

2.2. El uso del lenguaje como indicador de patrones de pensamiento

De acuerdo con Pennebaker, el lenguaje es un valioso indicador de rasgos, no solo de personalidad, sino también de situación social, económica y psicológica. En su libro *The Secret Life of Pronouns* [17] describe la investigación llevada a cabo por más de una década sobre el análisis del lenguaje y su relación con el estado interno de una persona. En este se señala que los pronombres, artículos, y preposiciones son las principales palabras que revelan en parte la personalidad, modo de pensar, estado emocional y conexiones con otros. Este conjunto de palabras, llamadas *palabras función* conforman el 1 % del vocabulario, y el 60 % del habla cotidiana.

Por otra parte, también menciona que se puede obtener una burda medida del estado emocional de un individuo al contar el uso que le da a palabras con connotación positiva y negativa. Parte de su estudio se basa en la creación de un sistema que cuenta con diccionarios asociados a estados mentales, tal sistema se describirá más adelante.

Bajo este esquema, se identificaron tres estilos de escritura, y por lo tanto, tres tipos de personalidades: formal, analítico y narrativo.

Por una parte, el estilo formal hace uso de palabras largas, uso numeroso de artículos, sustantivos, números y preposiciones. Las personas con este tipo de pensamiento tienden a preocuparse por su estatus y son menos auto-reflexivas. Por su parte, el estilo analítico se caracteriza por realizar distinciones, gran uso de palabras que indican negación, palabras adversativas y que cuantifiquen elementos. Las personas con este patrón predominante se desempeñan mejor en la escuela, son más honestas y están más abiertas nuevas experiencias. Por último,

se describe el estilo narrativo, que se caracteriza por el uso de lenguaje que involucra otras personas, uso de verbos en tiempo pretérito y lenguaje en el que predominan las conjunciones. Las personas que tienen un alto uso de este lenguaje tienden a tener mejores habilidades sociales, mayor número de amistades y se clasifican a sí mismas como más extrovertidas [17].

Esta investigación está fuertemente ligada a otras que estudian más minuciosamente las características individuales de la personalidad. Pueden apreciarse las relaciones que hay entre los rasgos generales marcados por el estilo de escritura y cada uno de los cinco rasgos de la personalidad definidas por el modelo Big-Five.

Con esto en mente, se plantea relacionar el uso coloquial del lenguaje en una plataforma informal, como Twitter, con los rasgos de personalidad de los autores del contenido en esa red social.

2.3. Aprendizaje automático

El aprendizaje automático es un campo multidisciplinario fuertemente influenciado por la inteligencia artificial, probabilidad y estadística, teoría de control, entre otros. Los algoritmos de aprendizaje automático usan datos históricos y aprenden de estos durante un periodo de entrenamiento, para entonces, predecir propiedades de datos desconocidos [11].

En el aprendizaje supervisado los datos incluyen atributos, o etiquetas, que se desean predecir. Un tipo de tarea que resuelve el aprendizaje supervisado, y en el que de hecho se enfoca este proyecto, es el de clasificación. En tareas de clasificación, el esquema de aprendizaje se presenta con un conjunto de datos etiquetados, de los cuales se espera obtener una manera de clasificar (o etiquetar) elementos que no han sido vistos por el algoritmo [11]. Al finalizar el entrenamiento, tendremos un modelo de clasificación. Así, se tiene que una función de aprendizaje $f : X \to Y$, donde X son las entradas o datos de los que se desea aprender, y Y corresponde a las salidas, o etiquetas a predecir. En la figura 2, se describe la secuencia general para generar un modelo de aprendizaje automático supervisado.

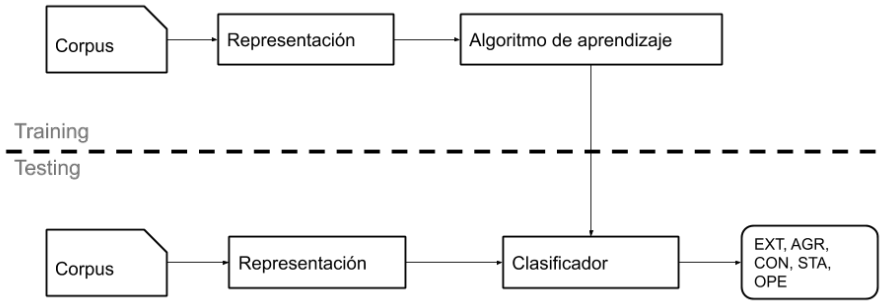

Figura 2: Proceso de generación de modelo de clasificación

El proceso para generar un modelo de clasificación consta de dos etapas: una de entrenamiento y otra de prueba. Durante la etapa de entrenamiento, se tiene un corpus de entrenamiento, a partir del cual se genera una representación. Esta es usada como entrada de un algoritmo de aprendizaje automático, lo que nos da como resultado un modelo clasificador. Así mismo, durante la etapa de prueba, se tiene una instancia de datos no vistos antes por el algoritmo, que son representados de la misma manera que en la etapa de entrenamiento. Esta representación se usa como entrada del clasificador, lo que da como resultado las etiquetas, o clases, correspondientes a la instancia de prueba.

Existen varios algoritmos de aprendizaje automático para generar un modelo de predicción. A continuación se explican brevemente los usados en este proyecto.

- *Árbol de decisión*: Este algoritmo usa un árbol de decisión como modelo predictivo, para navegar desde observaciones sobre un elemento (representado en las ramas), hasta las conclusiones sobre el valor final del elemento (representado en las hojas). El objetivo de estos algoritmos es generar un modelo que prediga el valor de un elemento aprendiendo reglas de decisión inferidas del conjunto de datos.

- *Bayes ingenuo multinomial*: Un clasificador de Bayes ingenuo pertenece a una familia de clasificadores probabilísticos basados en aplicar el teorema de Bayes, donde se asume independencia de los datos. En un modelo multinomial de Bayes ingenuo, las muestras de datos representan frecuencias en las que ciertos eventos han sido generados por una distribución multinomial $(p_1, ..., p_n)$, donde p_i es la probabilidad de ocurrencia de un evento i. Entonces, un vector de características $X = (x_1, ..., x_n)$ es un histograma, con x_i que cuenta el número de veces un evento i fue observado en una instancia particular. Este modelo es muy recurrido en clasificación de textos, donde los eventos representan la ocurrencia de una palabra en un documento.

- *Máquina de vectores de soporte*: Este conjunto de algoritmos representa los datos de entrada como puntos en un espacio, mapeados de tal manera que ejemplos de categorías separadas estén divididas por espacios grandes. Nuevas muestras de datos se mapean dentro del mismo espacio, y se clasifican dentro de una categoría basada en que parte de la división entre conjuntos de datos es la más cercana.

2.4. Representación de textos

La clasificación de textos o documentos es la tarea del aprendizaje automático supervisado que consiste en asignar categorías (o etiquetas) predefinidas a documentos de texto.

Con el fin de realizar la clasificación, los textos deben ser representados de manera sistemática y uniforme. A continuación se introducen las técnicas de representación usadas para este proyecto.

2.4.1. Bolsa de palabras

La manera más intuitiva de representar documentos de texto es mediante una bolsa de palabras.

En una bolsa de palabras, cada documento w es representado como un vector con n atributos.

$$w = [w_1, w_2, w_3, \cdots, w_n]$$

Donde w_i representa cada atributo en el documento. Los atributos corresponden a las palabras con ocurrencia en el documento. Las ocurrencias de cada palabra en cada texto en el conjunto de documentos, o corpus, establecen una relación término-documento. Esta relación término-documento se puede representar con una matriz que describe la ocurrencia de los términos en una colección de documentos. A esta matriz se le llama *matriz término-documento*. Los renglones de la matriz corresponden a documentos, mientras que las columnas corresponden a los términos.

A manera de ejemplo se tiene un par de documentos:

$d_1 =$ *Esta es una bolsa de palabras*,
$d_2 =$ *Esta bolsa de palabras tiene un pesado binario*

Esta relación término-documento se puede cuantificar de acuerdo a algún esquema de pesado. Algunos esquemas de pesado comunes son: binario, por frecuencia de términos (TF) y frecuencia de término por frecuencia inversa del documento (TF-IDF).

En el esquema de pesado binario se asigna a cada atributo un valor, ya sea 1 o 0, en función de si la palabra está o no en el documento.

Los atributos de los documentos d_1 y d_2, se pueden mapear a un diccionario:

['esta', 'es', 'una', 'bolsa', 'de', 'palabras', 'tiene', 'un', 'pesado', 'binario']

Y con base en este diccionario, se genera la representación como vector para ambos documentos:

$$d_1 = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]$$

$$d_2 = [1, 0, 0, 1, 1, 1, 1, 1, 1, 1]$$

La matriz término-documento correspondiente a estos dos documentos sería:

$$\begin{bmatrix} [1, 1, 1, 1, 1, 1, 0, 0, 0, 0] \\ [1, 0, 0, 1, 1, 1, 1, 1, 1, 1] \end{bmatrix}$$

Donde el primer renglón representa al documento d_1 y el segundo, representa al documento d_2. Y cada columna de la matriz representa cada término en el diccionario que representa los atributos de ambos documentos.

Donde cada atributo en ambos vectores corresponde a cada palabra en el diccionario. Cada 1 representa que la palabra está presente en el documento, en la posición que ocupa el atributo en el diccionario. Mientras que cada 0 indica que la palabra que ocupa ese lugar en el diccionario, está ausente del documento.

En el esquema de frecuencia de términos (TF), cada elemento en el arreglo corresponde a la cantidad de ocurrencias de cada atributo en el documento. A manera de ejemplo, se tiene un documento: *"Esta bolsa de palabras tiene un esquema de pesado diferente"*.

Así, su representación con el esquema de pesado por frecuencia de términos sería: [1, 1, 2, 1, 1, 1, 1, 1, 1]. Donde cada atributo del vector corresponde a la cantidad de veces que la palabra aparece en el documento.

En el esquema de pesado TF-IDF o Frecuencia de términos - frecuencia inversa de documento, se realiza una ponderación de la cantidad de veces que una palabra aparece en cada documento con respecto a su número de ocurrencias en todo el conjunto de documentos. Esto permite asignar un valor a cada atributo dependiendo de la importancia de la palabra en el texto o corpus [3].

2.4.2. Representaciones semánticas

Existe un modelo de lenguaje basado en predicciones, en el que se describen las probabilidades de ocurrencia de cada palabra. Estos modelos son estado del

arte en tareas como similitudes de palabras y analogías. Así tenemos la técnica conocida como Word2Vec [23], representaciones numéricas de las similitudes contextuales de palabras. Esto es, absorben información del contexto que rodea a las palabras. Word2Vec sirve a la finalidad de comprimir, en la representación de una palabra, la descripción más informativa posible para cada palabra. Este método debe entrenarse con una red neuronal para representar a cada palabra como un vector de atributos probabilísticos.

La representación de una palabra como un vector numérico se conoce como un *embedding* ("incrustación") de palabra. Con un embedding, se pretende mapear una palabra de un diccionario a un vector.

Word2Vec no es un algoritmo, sino una combinación de dos técnicas: Bolsa de palabras continua (CBOW) y modelos de Skip-gram [13]. Ambas técnicas mapean pesos para construir una representación de vectores.

La bolsa de palabras continua predice la probabilidad de ocurrencia de una palabra en un contexto. El contexto podría ser una palabra o un grupo de palabras.

Skip-gram presenta un esquema opuesto a la bolsa de palabras continua. Esto es, intenta predecir el contexto dada una palabra. El modelo de Skip-gram, si se configura con una palabra de distancia como en el CBOW, podrá predecir cuales son las dos palabras que rodean a una palabra.

Esta representación de palabras como vectores absorbe el contexto que rodea cada palabra y permite predecir la ocurrencia de una palabra en un contexto. Así, se tiene que palabras con vectores similares, se encuentran en contextos similares.

Dado que los embeddings de palabras representan numéricamente similitudes contextuales entre las palabras, se pueden manipular para realizar tareas como: encontrar la similitud entre dos palabras, encontrar palabras que no pertenecen a un contexto, o incluso, obtener la probabilidad de una oración dentro de un texto.

2.5. Métricas de evaluación

Las métricas de evaluación son usadas para determinar que tan adecuadamente un modelo desempeña su propósito. Ya sea de encontrar similitudes en grupos de elementos o, en el caso de este proyecto, clasificar documentos de texto de acuerdo a etiquetas correspondientes a rasgos de personalidad.

Una técnica para reportar los resultados de un clasificador es la matriz de confusión. Una matriz de confusión puede representarse mediante una tabla con las observaciones de los resultados de una clasificación. Las columnas en esta

tabla corresponden a las predicciones de clases y las filas corresponden a las clases reales. En la tabla 1, se describe una matriz de confusión.

	Clase positiva	Clase negativa
Predicción de clase positiva	Verdaderos positivos (TP)	Falsos positivos (FP)
Predicción de clase negativa	Falsos negativos (FN)	Verdaderos negativos (TN)

Tabla 1: Representación de matriz de confusión

Usando los valores obtenidos al intentar clasificar valores del corpus, se puede medir el desempeño de un algoritmo de aprendizaje con métricas de evaluación. Algunas métricas usadas para determinar la calidad de un clasificador son:

- *Precisión*: Es la razón entre en número de verdaderos positivos (TP) y de falsos positivos (FP):

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

  Intuitivamente, la precisión es la habilidad del clasificador de no etiquetar una muestra positiva como negativa.

- *Recuerdo*: La razón entre el número de Verdaderos positivos (TP) y de falsos negativos (FN):

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

  Intuitivamente, es la habilidad del clasificador para encontrar todas las muestras positivas.

- *F-Score*: Puede interpretarse como un promedio de la precisión (P) y el recuerdo (R):

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

- *ROC AUC*: Curva ROC, acrónimo de Recevier Operating Characteristic. Es la razón de verdaderos positivos frente a la razón de falsos positivos. Hace uso conjunto del recuerdo; una medida denominada especificidad o razón de verdaderos negativos (TNR), que se denota como la razón de verdaderos negativos (TN) y falsos positivos (FP):

$$TNR = \frac{TN}{TN + FP}$$

  Así también, hace uso de la razón de falsos positivos (FPR):

$$FPR = 1 - TNR = \frac{FP}{FP + TN}$$

De manera que una curva ROC se grafica con la razón de verdaderos positivos (TPR), también llamado recuerdo, contra la razón de falsos positivos (FPR), donde TPR está en el eje Y y el FPR está en el eje X La gráfica 3 muestra un ejemplo de esta curva.

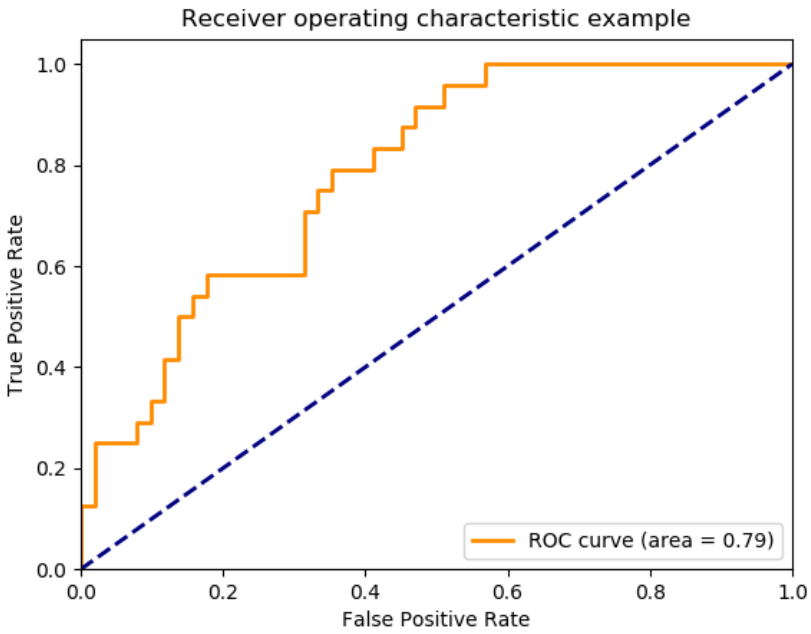

Figura 3: Gráfica de curva ROC. Fuente: https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics

2.6. Métricas de similitudes

Una métrica de similitud o distancia es una función $f(x, y)$ que define la distancia entre dos elementos, x y y, como un número real no negativo. Así, estas funciones permiten determinar que tan cercanos, o similares, son dos elementos. Estos elementos pueden ser números, vectores, matrices u algún otro objeto. Algunas métricas de distancia comúnmente usadas son: *distancia euclidiana*, *la similitud de cosenos*, y *la distancia de Manhattan*.

La *distancia euclidiana* representa el largo de una línea recta entre dos puntos en un espacio euclideano. Así, la distancia euclideana entre dos puntos p y q es el largo de la línea que los conecta ($\bar{pq}$).

Si te tienen dos puntos p y q en espacio euclidiano. Donde $p = (p_1, p_2, \cdots, p_n)$

y $q = (q_1, q_2, \cdots, q_n)$, su distancia euclidiana estaría definida como:

$$d_{euc}(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

La *similitud de cosenos* mide el coseno del ángulo entre dos vectores. En lugar de medir la distancia en un espacio, se mide la orientación de los puntos. Dos vectores con la misma orientación tienen una similitud de cosenos de 1, dos vectores con orientación de 90° entre sí, tienen una similitud de cosenos de 0 y dos vectores diametralmente opuestos tienen una similitud de cosenos de -1.

El cálculo de la similitud de cosenos está definido como:

$$d_{cos}(p, q) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \sqrt{\sum_{i=1}^{n} q_i^2}}$$

Para dos puntos p y q.

Finalmente, la *distancia de Manhattan* que se calcula con la suma de las diferencias en cada dimensión entre dos vectores. Esto traduce la cantidad de elementos diferentes entre dos vectores como un solo valor entero. A diferencia de las otras métricas que traducen la diferencia, o distancia, como un valor real.

La distancia de Manhattan está definida como:

$$d_{Man} = |x_1 - x_2| + |y_1 - y_2|$$

Para un elemento $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$.

2.7. Conceptos de desarrollo de aplicaciones web

Para visualizar los resultados y los datos necesarios, se desarrolló una aplicación web. La mayoría de los frameworks actuales están basados en una arquitectura Modelo-Vista-Controlador (MVC). La arquitectura MVC separa las representaciones internas de la visualización presentada al usuario. Una arquitectura Modelo-Vista-Controlador puede verse en la figura 4.

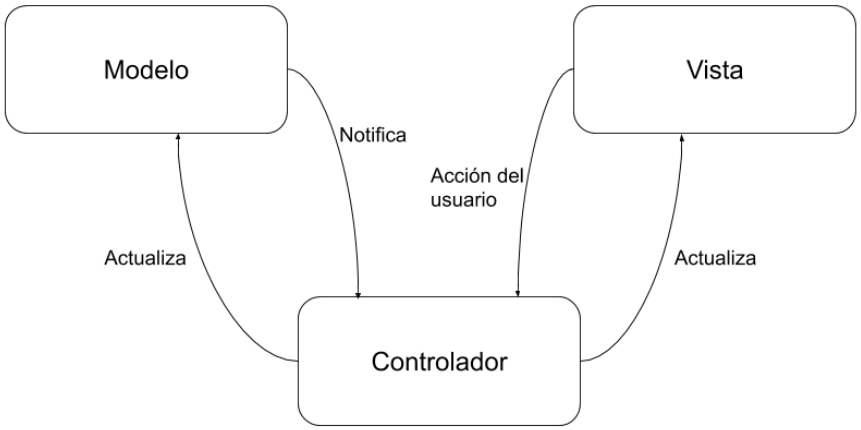

Figura 4: Esquema de arquitectura Modelo-Vista-Controlador

Los componentes de una arquitectura MVC son:

- *Modelos*: Son el componente principal de la arquitectura. Se trata de la estructura de la aplicación que maneja los datos, lógica y reglas de la aplicación.
- *Vistas*: La representación gráfica de la información, como gráficas, diagramas o tablas.
- *Controladores*: Expone la funcionalidad del sistema al usuario. Se encarga de aceptar entradas y convertirlas en comandos para los modelos y vistas.

Así, en una arquitectura Modelo-Vista-Controlador, el modelo es responsable de manejar los datos persistentes de la aplicación y recibe sus instrucciones de un controlador; la vista representa información de los modelos en formatos particulares; y el controlador responde a las entradas del usuario y solicita operaciones sobre los modelos de datos.

Ahora, una aplicación web debe exponer las diferentes páginas de las que se conforma. Las páginas de la aplicación pueden traducirse como acciones que el sistema puede realizar. Una técnica utilizada por los frameworks actuales para exponer los métodos del sistema a los usuarios es por medio de rutas. Las rutas son el medio por el cual una aplicación liga una URI con una acción. Así, se puede acceder a una funcionalidad de la aplicación, por ejemplo, el ingreso de un usuario a la aplicación por medio de una ruta: http://myapp.com/login. Donde http://myapp.com corresponde al nombre de la aplicación y todo lo que se encuentre después de éste, corresponde a la ruta que expone el método login.

Los resultados de los procesos en una aplicación web pueden ser desplegados en una vista. Las vistas pueden ser páginas estáticas en HTML o, como es común en frameworks web, pueden ser generadas de manera dinámica tomando datos, por ejemplo de modelos, y los convierte en una página HTML que es mostrada al usuario.

Los modelos por su parte, representan datos que se desea persistir para dar funcionalidad a la aplicación. Estos datos pueden estar almacenados en una base de datos para que los modelos los consuman. Generalmente, un modelo es instanciado como un objeto en el sistema, cuyos atributos pueden corresponder a los datos que se desea persistir. Por ejemplo los atributos del modelo de un usuario pueden ser su nombre, edad, género o cualquier otro atributo del que se desea guardar un registro permanente. Cuando se persiste un modelo, el framework de la aplicación se encarga de mapear los atributos del objeto a campos de la base de datos para guardarlos.

Los controladores, finalmente realizan las acciones que el usuario solicita. Por ejemplo, realizan consultas a la base de datos, que involucra crear una instancia de un modelo para operar sobre éste y, con sus datos, generar una vista para desplegarla al usuario.

El método para desarrollar la aplicación consistió en generar representaciones de corpus de texto para entrenar diferentes algoritmos de clasificación. Los modelos entrenados fueron evaluados para obtener el que mejor desempeño presenta para la tarea. Tras decidir por un esquema de representación y un modelo de clasificación, se desarrolló una aplicación web que integra estos procedimientos para consumir textos de un usuario en Twitter, y a partir de éstos, clasificar su personalidad de acuerdo con el modelo Big-Five.

3. Trabajo relacionado

Existen varias herramientas para evaluar a los usuarios en Twitter y extraer sus rasgos de personalidad. La mayoría de estas propuestas es producto de empresas privadas. Sin embargo, queremos hacer énfasis en las herramientas e investigaciones que han sido desarrolladas por investigadores involucrados en el estudio del lenguaje como indicador de patrones de pensamiento y personalidad.

3.1. Aplicaciones web

Pennebaker, fue el principal colaborador en el desarrollo del sistema LIWC-engine (Linguistic Inquiry and Word Count)[3]. Este sistema analiza textos en busca de palabras que reflejen características sociales y psicológicas de sus autores. LIWC-engine opera con base en diccionarios de categorías de palabras. Estas categorías engloban palabras relevantes para el perfil que se está analizando, ya sea social, psicológico o clínico [17]. Este sistema se empezó a desarrollar durante la década de los noventas y su primer versión se publicó en 2001. A partir de entonces ha recibido varias actualizaciones, principalmente a los diccionarios que le permiten el funcionamiento. La última de estas actualizaciones se realizó en 2015. Esta herramienta debe instalarse en la computadora que va a procesar los textos para funcionar, de manera que el uso del sistema requiere de su adquisición. Cabe destacar que LIWC-engine es una herramienta de paga.

Existe una herramienta online basada en LIWC, desarrollada en parte por el mismo Pennebaker, llamada Analyze Words[4]. Esta herramienta utiliza el motor de LIWC para analizar los tuits de un usuario registrado en Twitter, y luego despliega información gráfica sobre factores psicológicos y sociales del usuario examinado. Una limitante de este sistema es que, si bien permite visualizar los factores de personalidad de un usuario dentro del círculo social de otro, no permite hacer una comparación entre ambos.

Ahora, existen aplicaciones que extraen texto escrito de redes sociales para realizar perfilado de autor. Existe una aplicación desarrollada en 2016 que analiza los posts de un usuario en Twitter para obtener su perfil de edad, género y rasgos de personalidad [9]. Esto lo hace con técnicas de aprendizaje supervisado, con tal de validar las respuestas obtenidas a un cuestionario aplicado a los usuarios. Las respuestas de los usuarios se usan para entrenar modelos estadísticos de aprendizaje automático que puedan, posteriormente, producir resultados más acertados al examinar los escritos de usuarios en esta red social.

En contraste con los trabajos descritos anteriormente, el sistema que propone el presente trabajo basará su funcionamiento en el modelo Big-Five para hacer clasificación de la personalidad de los usuarios. También permitirá hacer

[3] http://liwc.wpengine.com
[4] http://www.analyzewords.com/

una comparación entre varios usuarios y sus factores de personalidad y luego desplegará esta información gráficamente al usuario mediante técnicas de representación de datos.

La tabla 2 realiza una comparativa entre las características de las herramientas descritas.

	Recopilación de textos de usuarios	Visualización gráfica de resultados	Basado en modelo Big-Five	Visualización de similitudes entre usuarios
LIWC (2015) [17]	Sí	No	No	No
Analyze Words [17]	Sí	Sí	No	No
Identifying your personality (2016) [9]	Sí	Sí	Sí	No
Herramienta propuesta	Sí	Sí	Sí	Sí

Tabla 2: Comparativa de herramientas relacionadas a la tarea con la herramienta propuesta para este proyecto

3.2. Trabajos de investigación

Los proyectos desarrolados por Pennebaker: LIWC y Analyze Words se basan en la misma investigación sobre psicología cognitiva. Esa investigación hace una descripción muy general de la personalidad basada en el uso que se le da al lenguaje. En el caso de Analyze Words, se clasifica el estilo de escritura de un usuario dentro de tres categorías: Estilo emocional, estilo social, y estilo de pensamiento. En el caso del estilo emocional, el sistema localiza cuatro tipos de palabras usadas: Enérgicas, de preocupación, de enojo y de lenguaje deprimido. Para el caso del estilo social, identifica cuatro tipos de vocabulario: conectado, personal, arrogante/distante y desconectado. Por último, para el estilo de pensamiento, encuentra tres tipos de vocabulario: Analítico, sensorial y en tiempo presente [17].

Se hace énfasis en que los sistemas basados en LIWC hacen clasificaciones muy generales a partir de un texto. Las características que extraen son muy diferentes a las indicadas en el modelo de Big-Five.

Otro trabajo más actual, *25 Tweets to Know You: A New Model to Predict Personality with Social Media* [4], publicado por IBM en 2017, se enfoca en reducir la cantidad de datos requeridos para modelar la personalidad de usuarios en Twitter con precisión. Este trabajo realiza una comparación con respecto a otros métodos y sistemas como LIWC. El resultado de este trabajo indica que su modelo de clasificación es más exacto que otras técnicas previas y requiere de ocho veces menos información de entrada para predecir la personalidad de un usuario a partir de su texto.

El método que propone combina embeddings de palabras con procesos Gaussianos. Extrae las palabras de los tuits de usuarios y promedia sus respectivas

representaciones de embeddings en un solo vector. Este vector se utiliza entonces para entrenar un algoritmo de aprendizaje supervisado. Específicamente, usaron el modelo GloVe de Twitter con vectores de 200 dimensiones y entrenado con 2 billones de tuits [18]. Como algoritmo de aprendizaje, usaron un modelo no lineal usado para regresión, conocido como proceso Gaussiano. Así, se usaron los vectores de embeddings como valores de entrada y se entrenó un modelo de proceso Gaussiano por cada una de los cinco rasgos del Big-Five.

La tabla 3 muestra los resultados obtenidos. Para evaluar el desempeño, usaron una correlación de Pearson entre los datos predichos y los valores verdaderos de los datos de entrenamiento. En una correlación de Pearson, o PCC, se mide la correlación lineal entre dos variables X y Y. El valor de una PCC, se mide entre 1 y -1, donde 1 es una correlación lineal positiva, 0 representa no correlación y -1 una correlación linear negativa. Es decir, mientras más se aproxime a 0, la correlación entre ambas variables es menor; un valor que se aproxime al 1, las variables están más relacionadas (ambas crecen o decrecen juntas en cantidades equivalentes); mientras que una puntuación que se aproxime al -1, una variable crece mientras la otra decrece una cantidad equivalente.

	Amabilidad	Responsabilidad	Extraversión	Neuroticismo	Apertura
PCC Reportado	0.29	0.33	0.25	0.42	0.37

Tabla 3: Resultados reportados del proyecto: *25 tuits to know you* [4]. El desempeño se midió con PCC.

Un trabajo de investigación reciente, *Texts for personality identification in undergraduates* [20], se enfocó en la predicción de personalidad de estudiantes universitarios. Ese proyecto generó un corpus con 418 textos relacionados a un perfil de personalidad.

Se usaron representaciones de n-gramas de palabras, n-gramas de caracteres y n-gramas de part of speech (POS). Estas se usaron para entrenar modelos con algoritmos de Bayes ingenuo, árbol de decisiones y máquina de vectores de soporte. Como métrica de evaluación para medir el desempeño en la tarea, usó el F-score.

En [20], se realizaron 405 experimentos (para cinco rasgos, nueve representaciones, tres esquemas de pesado, y tres algoritmos de aprendizaje). Los mejores resultados de esta investigación se presentan en la tabla 4.

	Configuración experimental		Score
	Representación	Clasificador	
Apertura	1-gram POS - TF	SVM	0.49
Responsabilidad	2-gram POS - TF	NB	0.39
Extraversión	5-gram Chars - Bool	DT	0.45
Amabilidad	1-gram words - TF	NB	0.45
Estabilidad emo.	1-gram POS - Bool	SVM	0.46

Tabla 4: Mejores resultados para clasificación de cada rasgo en trabajo de investigación TxPI-u [20]. Los resultados se muestran en F-Score.

Como puede verse, el desempeño para la tarea fue relativamente bajo, lo cual es muestra de la dificultad del problema de predicción de personalidad a partir de texto. El trabajo anterior propuso que técnicas nuevas de representación de textos u otras técnicas de aprendizaje podrían mejorar los resultados.

En la tabla 5 se muestra una comparativa de técnicas y datos generales de los trabajos de investigación que sirven como referencia a este proyecto.

Criterio	Herramientas		
	LIWC [17]	25 Tweets [4]	TxPI [20]
Idiomas	16 idiomas, incluyendo Inglés y Español	Inglés	Español
Esquema	Comparación de texto de entrada con diccionarios de emociones	Algoritmos de regresión entrenado con embeddings de palabras	3 Algoritmos de clasificación entrenados con bolsa de palabras
Datos	Texto de usuarios; diccionarios de palabras emoción	Embeddings GloVe entrenados con textos de 2B de tuits	Textos escritos por estudiantes; resultados de test de personalidad TIPI
Basado en modelo Big-Five	No	Sí	Sí

Tabla 5: Comparación de técnicas y datos usados en investigaciones previas con respecto a este proyecto.

4. Método propuesto

El esquema de trabajo general para desarrollar la herramienta. A partir de los textos de un usuario, se desea clasificar cinco atributos, cada una correspondiente a un rasgo del modelo Big-Five. Esto se hace tanto para el usuario que solicita el análisis, como para los amigos de este usuario. Las predicciones de estas clases conforman un *vector de personalidad*.

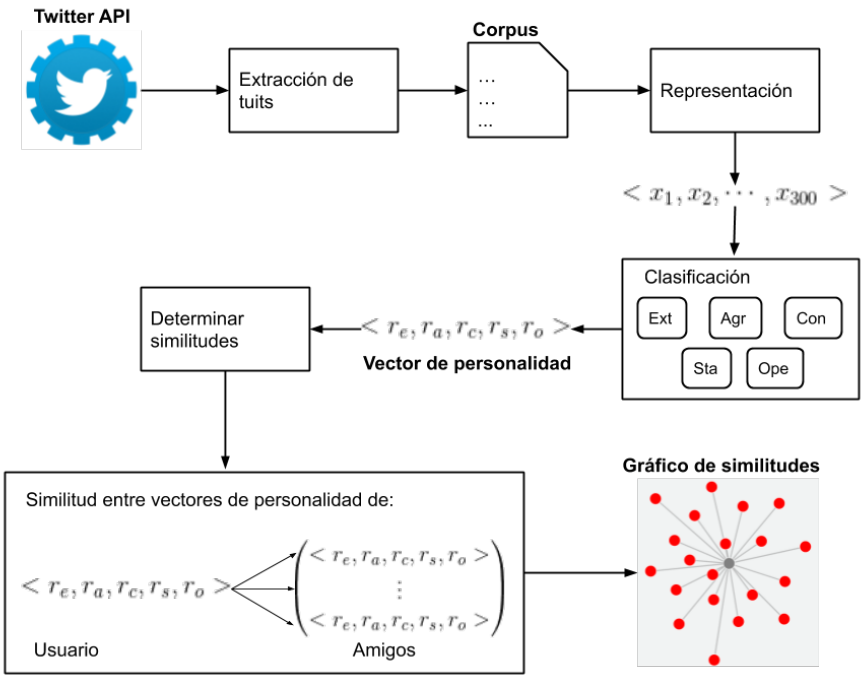

Figura 5: Framework general del método propuesto.

En la figura 5, se muestra el framework general del método propuesto. Las diferentes etapas del método se explican a continuación:

Para iniciar la tarea, es necesario extraer contenido de Twitter. La API de esta red social expone los métodos necesarios para consumir información de los usuarios: nombres, lista de publicaciones, amigos, etc. Las publicaciones de un usuario son extraídas y recopiladas como un solo conjunto de textos, conocido como corpus. En la etapa de representación, el corpus correspondiente a los textos de un usuario se representan de manera numérica. Esta representación se alimenta a los modelos entrenados para la tarea en la etapa de clasificación. Cada uno de los modelos determina el valor para cada rasgo del modelo Big-Five, este valor puede ser ya sea uno o cero. El resultado de la etapa de clasificación

es un *vector de personalidad*. Este vector está formado por cinco valores, cada uno correspondiente a un rasgo de personalidad.

$$C(x) = [r_e, r_a, r_c, r_s, r_o | r : (1, 0)]$$

Así, un clasificador C recibe una instancia de entrada x, que corresponde al *vector de representación* de la que se desea obtener etiquetas. Con esa entrada, se genera el *vector de personalidad* con cinco etiquetas r. Cada una de las etiquetas corresponde a un rasgo del modelo Big-Five: Extraversión, Amabilidad, Responsabilidad, Estabilidad emocional y Apertura, en ese orden. Cada etiqueta puede tener un valor ya sea uno o cero, que corresponden una polaridad positiva o negativa, respectivamente, para cada rasgo.

Tras obtener el *vector de personalidad* de un usuario, es necesario un método para compararlo con el de otros usuarios. Varias técnicas fueron evaluadas para obtener la representación más significativa de las diferencias (o similitudes) entre usuarios. Fueron tomadas en cuenta: *distancia de Manhattan, similitud de cosenos y distancia euclidiana* que se describen en la sección 2.6. Se compararon estas métricas para decidir por una que represente mejor las similitudes entre los *vectores de personalidad* de los usuarios. La tabla 6 muestra la comparación entre un usuario (*vector de personalidad* 1) contra el de otros usuarios (*vector de personalidad* 2). El primer vector de personalidad comparado tiene un rasgo diferente con respecto al del usuario; el segundo contiene dos rasgos diferentes, el tercero es exactamente igual al del usuario y el cuarto es completamente opuesto

Vector de personalidad 1	Vector de personalidad 2	Métricas		
		Distancia Euclidiana	Similitud de cosenos	Distancia de Manhattan
[1, 1, 1, 1, 0]	[1, 1, 1, 1, 1]	1.0	0.89	1
”	[1, 1, 1, 0, 1]	1.41	0.75	2
”	[1, 1, 1, 1, 0]	0.0	1.0	0
”	[0, 0, 0, 0, 1]	2.23	0.0	5

Tabla 6: Comparación de métricas de similitud.

Tanto la distancia Euclidiana como la similitud de cosenos despliegan la similitud o diferencia entre vectores de personalidad con un número real. Por su parte, la distancia de Manhattan corresponde a la cantidad de elementos diferentes entre dos vectores. De esta manera, se tiene que la distancia de Manhattan destaca información que es más relevante a este proyecto, a saber, la diferencia entre dos vectores de personalidad. Por lo tanto, se decidió usar la distancia de Manhattan como métrica de similitud a usar en la herramienta.

La comparación entre dos vectores de personalidad permite determinar un nivel de afinidad, o similitud, entre un usuario y sus amigos. Este nivel de afinidad es presentado al usuario de una manera gráfica. Las métricas de distancia permiten representar esta afinidad. Esto es, una distancia menor entre dos vectores, implica una mayor afinidad, mientras que una distancia mayor equivale a afinidad muy baja.

Se realizó un trabajo experimental para validar el esquema del método propuesto. En este trabajo experimental se desea encontrar una combinación de esquema de representación y modelo de aprendizaje que se desempeñara mejor para la tarea. En la siguientes secciones se describirá el procedimiento experimental realizado, así como los recursos de datos que se usaron para generar un clasificador de rasgos basado en el modelo Big-Five.

4.1. Conjunto de datos

El corpus usado en esta tarea consiste en un conjunto de instancias de texto de estudiantes universitarios de diversas carreras. Este corpus fue usado en la tarea *Hand written texts for personality identification* [19].

El corpus usado está dividido en dos particiones: entrenamiento y prueba. La tabla 7 muestra algunas características del corpus.

Partición	Instancias de texto	Vocabulario promedio
Train	418	67.85
Test	125	64.65

Tabla 7: Distribución y vocabulario promedio de particiones del corpus

Cada instancia está asociada a un archivo de texto. El vocabulario promedio se refiere a la cantidad de palabras usadas en cada instancia de archivo.

Para elaborar este corpus, se obtuvo el perfil de personalidad de los participantes aplicando un instrumento de evaluación. El instrumento usado, debido a su rapidez de aplicación, fue el TIPI [21] (Ten Item Personality Inventory) en su versión en español. Luego se le pidió a los participantes escribir un texto sobre experiencias personales.

De esta manera se tienen los perfiles de personalidad de acuerdo con el modelo Big Five y, cada uno, asociado a una muestra de texto de los participantes.

Sobre la asignación de perfiles de personalidad, los textos se encuentran etiquetados con los cinco rasgos. Cada uno de los rasgos se mapea a dos polaridades: positiva y negativa. La tabla 8 muestra el número de individuos por clase de cada rasgo de personalidad.

	Train		Test	
	Polaridad positiva	Polaridad negativa	Polaridad positiva	Polaridad negativa
Apertura	239	179	71	54
Responsabilidad	171	247	61	64
Extroversión	212	206	68	57
Amabilidad	177	241	61	64
Estabilidad emo.	186	232	73	52

Tabla 8: Número de participantes que presentan cada polaridad de rasgos del Big-Five

El corpus de textos viene acompañado de un archivo que contiene las etiquetas de cada instancia. Estas etiquetas corresponden a la polaridad de rasgos de Big-Five, ordenadas en un renglón por archivo. La tabla 9 muestra un ejemplo de la relación entre las instancias de texto con su contraparte en el archivo de etiquetas.

Archivo de texto	Archivo de etiquetas				
	Ext	Agr	Con	Sta	Ope
DSC_0085_hxpi.txt	0	1	1	0	0
DSC_0084_hxpi.txt	1	1	1	0	0
DSC_0087_hxpi.txt	1	0	0	1	0

Tabla 9: Ejemplo de relación entre los archivos individuales de texto de participantes y sus respectivas etiquetas de polaridad

Este corpus fue usado para entrenar y evaluar los modelos para clasificar textos de acuerdo a perfiles de personalidad.

4.2. Archivo de embeddings

En este proyecto se usó una representación de documentos de texto con embeddings de palabras entrenados con un algoritmo Word2Vec. Se utilizó un archivo de embeddings con textos en español, entrenado con una muestra de cerca dos millones de palabras; con cada palabra representada como un vector de trescientos elementos. El proyecto que construyó los embeddings fue llevado a cabo en conjunto con el Laboratorio de Tecnologías del Lenguaje de Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Se espera que, al representar los textos como una ponderación de embeddings de palabras, los vectores obtenidos representen la presencia o ausencia de ciertos rasgos de manera más eficiente que otras representaciones.

El archivo de embeddings, que se usó para este proyecto, es de tamaño considerable (aproximadamente 6Gb) y produjo algunos percances que se analizarán posteriormente en el documento.

4.2.1. Consideraciones para validar el uso del archivo de embeddings

Para validar que representar el corpus con los embeddigs Word2Vec sea viable, se comprobó que las palabras existentes en el corpus estuvieran presentes en el archivo de embeddings. De la misma manera, se midió la presencia de palabras relacionadas a cada polaridad de los rasgos del Big Five. Lo anterior para detectar deficiencias y predecir correctamente las etiquetas de cada rasgo.

Como análisis preliminar, se obtuvo el porcentaje de palabras del corpus con presencia en el archivo de embeddings. Esta cobertura fue del 88.71 %. Así, se consideró que era viable usar el archivo de embeddings como fuente para representar los textos del corpus. De igual modo, se obtuvo la cobertura de palabras relacionadas a cada polaridad de rasgos. Es decir, las palabras relacionadas a cada instancia etiquetada con cierta polaridad en cada rasgo.

La tabla 10 muestra la cobertura de palabras correspondientes a polaridades de rasgos en el archivo de embeddings.

	Positiva	**Negativa**
Extroversión	91.27 %	90.09 %
Amabilidad	90.36 %	91.03 %
Responsabilidad	90.85 %	90.46 %
Estabilidad	92.35 %	89.36 %
Apertura	92.65 %	88.95 %

Tabla 10: Cobertura de textos en embeddings correspondientes a cada polaridad de rasgos.

Con el análisis anterior, se considera viable usar la representación basada en embeddings Word2Vec para entrenar modelos que identifiquen rasgos del modelo Big-Five.

4.3. Representación de documentos con esquema de embeddings Word2Vec

Para representar un documento usando embeddings, es necesario realizar un procedimiento especial. Para esto, se hace uso del archivo descrito en la sección 4.2, donde se tiene un conjunto de cerca de dos millones de palabras, cada una representada con un vector de trescientos elementos.

Se deben extraer los embeddings individuales de cada palabra del documento.

$$D = [w_1, w_2, \cdots, w_n]$$

Así, se dice que un documento D está conformado por una cantidad n de palabras w. Cada una de estas palabras tiene a su vez una representación de embeddings.

$$w = [x_1, \cdots, x_{300} | x : \mathbb{R}]$$

Donde una palabra w se representa como un vector de trescientos elementos numéricos reales x.

La representación completa de un documento podría expresarse con una matriz.

$$D = \begin{bmatrix} [d_1 w_1, d_1 w_2, \cdots, d_1 w_{300}] \\ [d_2 w_1, d_2 w_2, \cdots, d_2 w_{300}] \\ \vdots \\ [d_n w_1, d_n w_2, \cdots, d_n w_{300}] \end{bmatrix}$$

Donde d_i corresponde a cada documento, w_i corresponde a cada palabra en ese documento.

Así, un documento completo es representado como una matriz donde cada renglón corresponde a cada uno de los embeddings que representan cada palabra en el documento, mientras que cada columna corresponde a un valor numérico en una posición dada del vector que representa la palabra.

Ahora, para representar un documento completo como un solo vector, es necesario combinar todos los vectores de embeddings de palabras (renglones) en uno solo. Para lograr esto, se hizo un promedio de cada columna, que corresponde a un valor numérico, de la matriz.

$$D = \left[x_1, x_2, \cdots, x_{300} | x_j = \frac{\sum_{i=1}^{n} d_i w_j}{n} \right]$$

Donde d_i corresponde a cada documento, w_j representa cada palabra del documento d_i, en una colección de documentos n.

Cada uno de los nuevos elementos del vector que representa al documento D corresponde al promedio de todos los valores numéricos que ocupan la misma columna. De esta manera, un solo documento queda representado como un vector de trescientos elementos numéricos reales.

4.4. Baseline

El baseline para la tarea se describe en la competencia *Multimedia Information Processing for Personality & Social Networks Analysis* [19], en su tarea

de clasificación de texto. El objetivo de esta tarea fue el de predecir los rasgos de personalidad de usuarios a partir de texto escrito. Se usó el mismo conjunto de datos que el usado en este proyecto: una partición de entrenamiento de 418 ensayos, cada uno asociado a clase 1 o 0, correspondientes a polaridad alta o baja de cada rasgo de personalidad; y una partición de prueba con 125 ensayos, etiquetados de la misma manera.

En el baseline, se representaron los textos usando 3-grams de caracteres con pesado TF de una bolsa de palabras, y como algoritmo de aprendizaje, se usó una Máquina de Vectores de Soporte. Los resultados obtenidos se muestran en la tabla 11. Se usó la métrica de Área Bajo la Curva ROC para medir el desempeño de los métodos de clasificación.

Rasgo	ROC AUC
Apertura	0.545
Responsabilidad	0.5
Extroversión	0.543
Amabilidad	0.463
Estabilidad emo.	0.606

Tabla 11: Resultados de clasificación de clase por rasgo del baseline.

4.5. Proceso de experimentación

Se creó un módulo en Python que realiza las operaciones necesarias para preprocesar corpus, generar representaciones de éste, entrenar modelos de clasificación y realizar las pruebas con los clasificadores. Esto nos permite compararlos y decidir cuál presenta mejor desempeño.

El proceso general de experimentación se muestra en la figura 6.

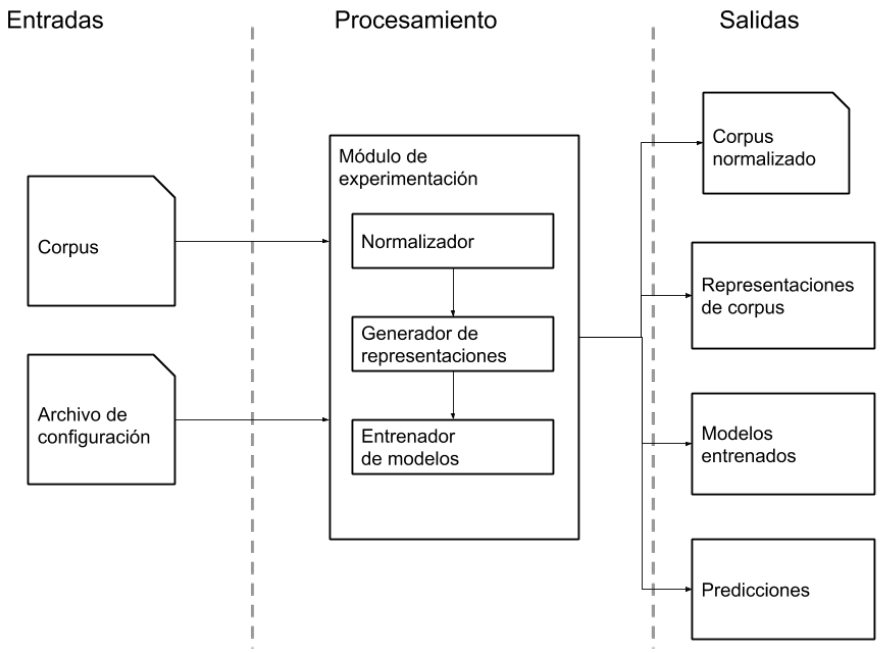

Figura 6: Proceso general de experimentos.

Como entradas para el módulo de experimentación, se tiene el corpus, ya sea de entrenamiento o de prueba, que consiste en el conjunto de archivos de HWxPI con los que se entrenarán los modelos. también está el archivo de configuración, donde se especifican los preprocesamientos a realizarle al corpus; las representaciones del corpus que se desea generar; y los modelos que se desea entrenar con estas representaciones.

Durante la etapa de procesamiento, el módulo principal invoca a sub-módulos que normalizan el corpus, generan representaciones de éste y se entrenan modelos de clasificación con las representaciones generadas.

A continuación se describen brevemente los procesos individuales que se realizan en la experimentación.

Para la normalización, se toma en cuenta que el corpus HWxPI contiene etiquetas para describir fenómenos del lenguaje. La normalización hecha para este proyecto, se dedica a eliminar estas etiquetas del corpus preprocesado.

Para representar el corpus, se creó un sub-módulo generador de representaciones. Este módulo toma cada archivo del corpus preprocesado en la etapa anterior y genera la matriz término-documento del corpus. La figura 7 muestra el proceso general de representación realizado al corpus para el proyecto.

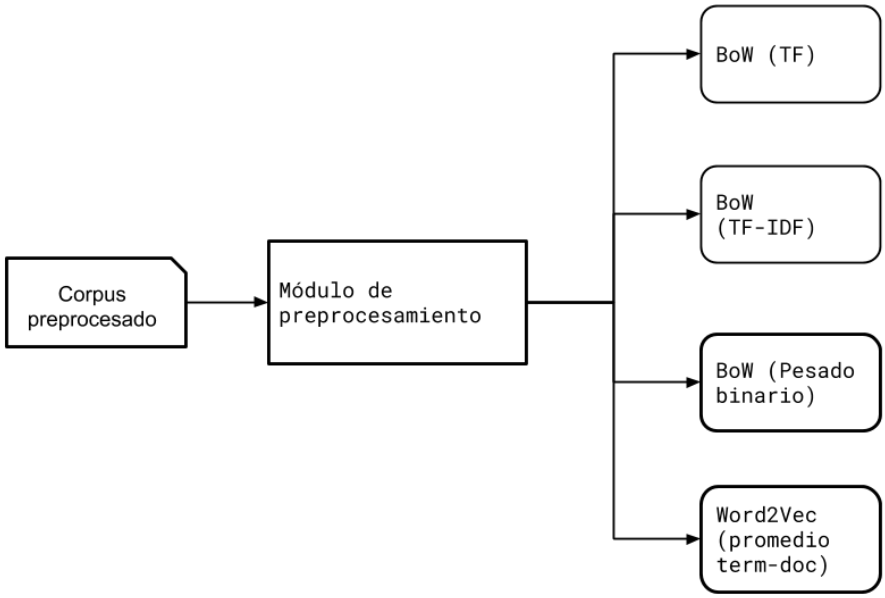

Figura 7: Módulo de representación que recibe el corpus preprocesado y genera las representaciones

Las representaciones que es posible realizar con este módulo son tres versiones de pesado de bolsa de palabras, binario, TF y TF-IDF; así como una representación de vectores de embeddigs Word2Vec. Esto implica que potencialmente se podrán tener cuatro representaciones del mismo corpus.

Para entrenar los modelos de clasificación, se creó un sub-módulo entrenador de modelos, expresado en la figura 8. Este módulo recibe iterativamente cada representación generada en la etapa anterior, con la cual entrena a cada uno de los algoritmos seleccionados.

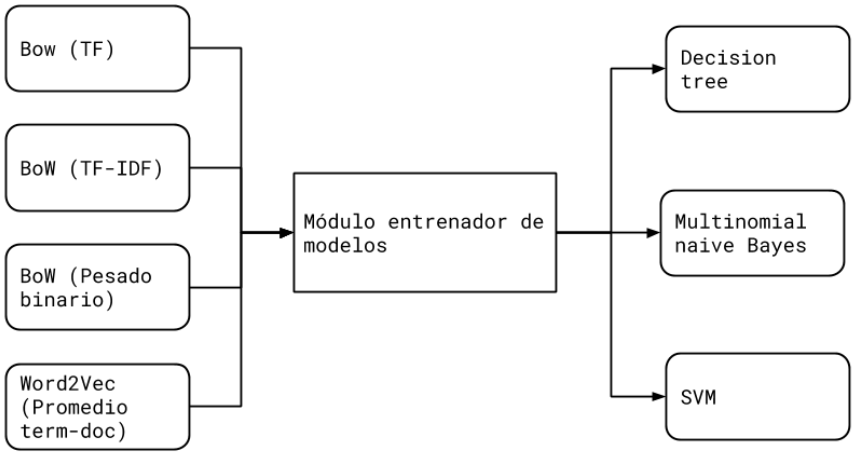

Figura 8: Módulo de entrenamientos. Cada representación generada en la etapa anterior es usada como entrada para los algoritmos de entrenamiento.

Los algoritmos de aprendizaje automático usados en esta etapa son: Árbol de decisión, Bayes ingenuo Multinomial y Máquina de vectores de soporte.

Las implementaciones de algoritmos de aprendizaje supervisado son: *svm* para una máquina de vectores de soporte (Support Vector Machine); *tree* para algoritmos de árboles de decisión; y finalmente, *MultinomialNB* para un algoritmo de Bayes Ingenuo Multinomial.

La partición de prueba del corpus es preprocesada exactamente igual que la partición de entrenamiento. Igualmente, se realizan las mismas representaciones aplicadas a la partición de entrenamiento. Los modelos construidos en la etapa de entrenamiento son usados como entrada nueva del módulo de entrenamientos. Se predicen con estos modelos las etiquetas de la partición de prueba del corpus.

4.6. Experimentos y resultados

Se requiere predecir la polaridad, ya sea positiva o negativa, de cada rasgo del modelo Big-Five. Por lo tanto, se abordan cinco problemas de clasificación, cada uno de tipo binario. Se realizaron dos conjuntos de experimentos, uno por cada clase de representación del corpus: una representación basada en bolsa de palabras y otra basada en embeddings Word2Vec. Para las representaciones del corpus basadas en bolsa de palabras, se usó el pesado binario, TF y TF-IDF. Para la representación basada en embeddings Word2Vec se realizó el procedimiento descrito en la sección 4.3. Se entrenaron tres algoritmos de aprendizaje supervisado: Árbol de decisión, Bayes Ingenuo Multinomial y Máquina de Vec-

tores de Soporte. Los utilizó la métrica ROC AUC para evaluar el desempeño de los clasificadores.

Para el primer conjunto de experimentos, se entrenaron los tres algoritmos de aprendizaje con los tres pesados de bolsa de palabras mencionados: binario, TF y TF-IDF. Los resultados obtenidos para estos experimentos, se muestran en la tabla 12.

	Bolsa de Palabras								
	Árbol de decisión			Bayes Ingenuo Multinomial			Máquina de Vectores de Soporte		
	Bool	TF	TF-IDF	Bool	TF	TF-IDF	Bool	TF	TF-IDF
Apertura	**0.484**	0.449	0.516	0.482	0.477	0.5	0.479	0.451	0.482
Responsabilidad	**0.555**	0.504	0.502	0.472	0.472	0.5	0.530	0.513	0.471
Extraversión	0.504	0.441	0.478	0.473	0.516	0.507	0.497	0.498	**0.540**
Amabilidad	0.463	**0.504**	0.446	0.438	0.464	**0.504**	0.455	0.479	0.462
Estabilidad emo.	0.481	0.521	**0.550**	0.492	0.546	0.5	0.493	0.511	0.529

Tabla 12: Resultados para primer conjunto de experimentos: Bolsa de palabras. Puntuaciones en ROC AUC. Mejor puntuación por rasgo en negritas.

Para el segundo conjunto de experimentos, los tres algoritmos de aprendizaje se entrenaron con una representación basada en embeddings Word2Vec. La tabla 13 muestra los resultados para este conjunto de experimentos en ROC AUC.

	Embeddings Word2Vec	
	Árbol de decisión	Máquina de Vectores de Soporte
Apertura	**0.545**	0.486
Responsabilidad	0.415	**0.572**
Extraversión	0.457	**0.543**
Amabilidad	0.504	**0.511**
Estabilidad emo.	**0.533**	0.489

Tabla 13: Resultados para segundo conjunto de experimentos: embeddings Word2Vec. Puntuaciones en ROC AUC. Mejor puntuación por rasgo en negritas.

En la tabla 14, se hace una comparación entre los mejores resultados de cada conjunto de experimentos.

Mejores resultados				
	Bolsa de Palabras		Word2Vec	
	Configuración	Score	Configuración	Score
Apertura	Bool - DT	0.484	DT	**0.545**
Responsabilidad	Bool - DT	0.555	SVM	**0.572**
Extraversión	TF-IDF - SVM	0.540	SVM	**0.543**
Amabilidad	TF - DT	0.504	SMV	**0.511**
Estabilidad emo.	TF-IDF - DT	**0.550**	DT	0.533

Tabla 14: Comparative de los mejores resultados entre los dos conjuntos de experimentos. Puntuaciones en ROC AUC. Mejores resultados por rasgo en negrita.

4.7. Discusión

Como puede verse en la tabla 14, una representación basada en embeddings Word2Vec se desempeñó mejor que una basada en bolsa de palabras para cada rasgo, excepto para predecir *estabilidad emocional*. Así también, se ve que el modelo entrenado con algoritmo de máquina de vectores de soporte se desempeñó ligeramente mejor en tres de los cinco rasgos: *Responsabilidad*, *Extraversión* y *Amabilidad*. Por este motivo, se eligió el modelo entrenado con clasificador de vectores de soporte y con representación de embeddings Word2Vec para ser usado en este proyecto.

Dado que el trabajo de baseline y el presente proyecto usaron el mismo conjunto de datos y métrica de evaluación, se puede realizar una comparación entre los resultados de ambos proyectos. La tabla 15 hace esta comparación entre los mejores resultados obtenidos en este proyecto y los resultados de la competencia [19], en su tarea HWxPI.

	HWxPI	Experimentos en este proyecto
	ROC AUC	ROC AUC
Apertura	0.545	0.545
Responsabilidad	0.5	0.572
Extroversión	0.543	0.543
Amabilidad	0.463	0.511
Estabilidad emo.	0.606	0.533
Promedio	0.531	0.54

Tabla 15: Comparación entre puntuaciones de la tarea HWxPI contra los obtenidos en los experiementos de este proyecto.

Al comparar Las puntuaciones de los experimentos en este proyecto con las del baseline, se encontró que se mejoraron ligeramente para ciertos rasgos

como *Responsabilidad* y *Amabilidad*; mientras que para el caso de *Apertura* y *Extroversión*, las puntuaciones se igualaron. La única puntuación que no se pudo igualar con respecto al baseline, fue *Estabilidad emocional*. En cuanto al promedio de todas las puntuaciones de predicción de rasgos, el obtenido en este proyecto fue ligeramente mejor al del baseline.

Identificar la personalidad de un usuario con base en una muestra corta de texto es una tarea complicada. Esto queda de manifiesto en las bajas puntuaciones obtenidas en los experimentos.

5. Desarrollo de la aplicación

5.1. Descripción de la aplicación

Una vez concluida la etapa de experimentación y habiendo seleccionado un modelo para predecir rasgos de personalidad, se procedió a implementar una aplicación web. La herramienta tiene como funciones: la descarga de textos (tuits) de un usuario; la predicción de los rasgos de personalidad del usuario con base en los textos obtenidos; la predicción de los rasgos de personalidad de los amigos del usuario (cuentas en Twitter que el usuario sigue). La arquitectura general del sistema se muestra en la figura 9.

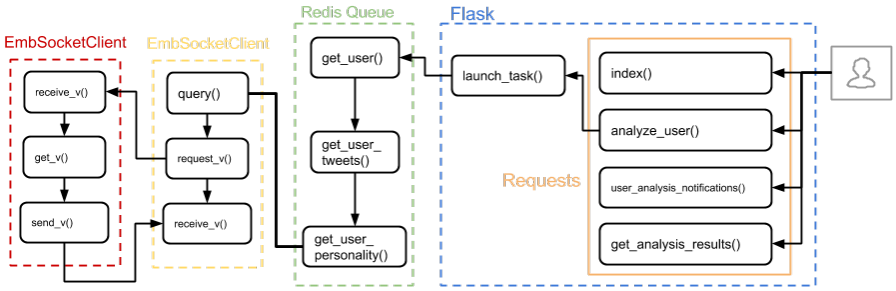

Figura 9: Arquitectura general de la aplicación

Se observa en el cuadro azul, el envoltorio principal de la aplicación en Flask. Dentro de este, en naranja, los diferentes enpoints o requests a los que puede acceder el usuario. Se tiene la cola de Redis en verde, con los procesos que realiza para analizar un usuario. Existen módulos, que se describirán en detalle en las siguientes secciones, específicamente para la tarea de representación de textos como vectores. El módulo *EmbSocketClient*, se alimenta del módulo *EmbSocketServer*. Este último tiene cargado el archivo de embeddings y puede consultar de este los vectores de palabras, mientras que el primero hace peticiones por el embedding de una palabra específica.

A continuación se describen los componentes y procesos llevados a cabo en el diagrama 9.

En el recuadro azul de la figura 9, se muestra el contenedor de la aplicación. Las funciones principales se muestran en el recuadro naranja, y corresponden con los endpoints o métodos que la aplicación expone al usuario. El método *index* es usado para exponer la página de inicio de la aplicación. El método *analize_user* es el principal y se encarga de invocar al método *launch_task*, para desencadenar los procesos necesarios para analizar un usuario, se explicarán estos procesos posteriormente. El método *user_analysis_notifications* se usa pa-

ra enviar notificaciones al usuario sobre el progreso del análisis que solicitó. El método *get_analysis_results* muestra los resultados del análisis al usuario.

Los recuadros rojo y amarillo contienen los módulos cliente y servidor para la tarea de consulta a los embeddings de palabras. Estos módulos fueron creados debido a dificultades sobre al carga del archivo de embeddings. Los detalles de estos módulos se explicarán en la sección 5.3

5.2. Tecnologias usadas en la aplicación

El sistema fue desarrollado usando el framework *Flask* de *Python*. El framework está basado en una arquitectura basada en rutas, vistas y modelos. Es necesario definir *endpoints*, las rutas que accederán los usuarios para conectarse y utilizar las funciones del sistema; vistas, que muestran interfaces al usuario; y modelos, que definen la estructura de entidades de las que se desea guardar un registro permanente.

Para solicitar el proceso de análisis, se accede a la ruta *Analyze User* de la aplicación. La función asociada con esta ruta obtiene los valores necesarios para desplegarlos en la vista. El proceso bloquea el request, es decir, la página permanece en proceso de carga hasta que la operación se complete. En una aplicación convencional, este proceso involucra quizá solo consultar la base de datos o realizar alguna operación sencilla. Para el caso de este proyecto, sin embargo, el proceso involucra también hacer llamadas a la API de Twitter, realizar las representaciones necesarias y hacer las predicciones de rasgos. Por lo anterior, el request permanece bloqueado hasta que el análisis se complete. Esta tarea puede llevar varios minutos. Era necesario realizar el proceso en background para no bloquear el request. Así, se tiene la posibilidad de desplegar otra página al usuario que muestre el progreso de su análisis.

Con ese propósito, se utilizó el manejador de tareas asíncronas *Redis Queue*. Este puede encolar las tareas realizadas en background, y expone métodos que permiten consultar su progreso, estado, resultados, etc.

Redis asigna identificadores a cada proceso en su cola de procesos. Este id fue usado también para representar a las entidades de los procesos en la base de datos.

Para llevar a cabo la tarea de descarga de tuits de usuarios, se utiliza la biblioteca *Twython* [5]. Esta biblioteca hace llamadas a la API de Twitter para obtener información como: nombre del usuario, número de tuits, número y listado de amigos, así como el conjunto de tuits del usuario.

Para realizar la representación de los textos como vectores y consultar los

[5] https://twython.readthedocs.io/en/latest/

embeddings de palabras en los tuits del usuario, se usa la biblioteca *gensim* [6].

Por último se usa la biblioteca *Scikit-Learn* [7] para realizar las predicciones de las etiquetas correspondientes a rasgos de personalidad de cada usuario.

A continuación, se presentan brevemente algunos componentes del sistema que forman parte de la aplicación *Flask*.

5.2.1. Modelos

Los modelos representan la estructura de las entidades en la base de datos. Entidades como los usuarios tendrían lugar como un modelo. La implementación de la aplicación no requiere guardar registro permanente de los visitantes, por lo tanto no se almacenan datos sobre los usuarios. Entidades que sí están presentes en el sistema son los procesos que se llevan a cabo y los resultados obtenidos de estos procesos.

Los dos modelos presentes en el sistema son los siguientes.

Task		
Campo	**Tipo**	**Descripción**
Id	String	Cadena identificadora del proceso. Se utiliza identificador asignado por la cola de Redis.
Name	String	Nombre de la tarea en ejecución.
Description	String	Descripción de la tarea en ejecución.
Complete	Bool	Bandera que indica si la tarea está en proceso o finalizada.
Complete_with_status	Int	Código indica condiciones de finalización de la tarea.
user_ip	String	Ip del usuario que solicita análisis.

Tabla 16: Modelo que representa tareas de análisis en el sistema

[6] https://radimrehurek.com/gensim/
[7] https://scikit-learn.org

Result		
Campo	**Tipo**	**Descripción**
Id	String	Cadena identificadora del proceso. Se utiliza identificador asignado por la cola de Redis.
Result	Text	Resultado de la ejecución. Se almacena una estructura json con un vector por usuario.

Tabla 17: Modelo que representa resultados de análisis de usuario

5.2.2. Rutas

Las rutas son los endpoints de la aplicación. Funciones que pueden ser accedidas por medio de una url. En un navegador se puede acceder a ellas con la url de la aplicación, seguido del endpoint al que se desea acceder. Las funciones ruta pueden ejecutar tareas como consultar la base de datos, hacer operaciones, etc. Las funciones ruta retornan, por los general, una vista que se arma con los datos necesarios. Las rutas del sistema son las siguientes.

- *Index*: Es accedida con un endpoint raiz ('/') o directamente ('/index'). Puede recibir el identificador en Twitter de un usuario, que es entonces transferido a la ruta de análisis para proceder con la tarea. Retorna la vista *index.html*.
- *Analyze user*: Se accede con el endpoint '/analyze_user/'. Recibe la cadena identificadora del usuario en Twitter. Se valida que el usuario no tenga un análisis en ejecución y lanza la tarea de análisis del usuario. Retorna la vista *useranalysis*.
- *User analysis notifications*: Ruta que alimenta el progreso de la vista *useranalysis*. Consulta el progreso de la tarea de análisis del usuario en la base de datos. Retorna una estructura json con el porcentaje de progreso de la tarea y la descripción del estado del proceso actual.

5.2.3. Vistas

Las vistas representan las páginas visibles para el usuario. Estas son contruidas con HTML simple y el framework generador de vistas de Flask: *jinja*. Las vistas presentes en el sistema son:

- *base.html*: Vista template del sistema. En esta, se definen elementos comunes para la aplicación, importación de estilos, scripts, etc. Otras vistas heredan todos los elementos de esta.

- *index.html*: Vista inicial. La página de bienvenida de la aplicación. Se presenta un formulario sencillo, donde el usuario puede ingresar su identificador en Twitter para comenzar el análisis.

- *useranalysis.html*: Esta vista se despliega después de que el usuario solicite el análisis. En esta vista, se despliega una barra de avance para indicar el progreso del análisis. Se incluye un texto que sirve de descripción sobre la tarea que se está realizando en el momento. Tanto la barra de avance como el texto de descripción se alimentan con llamadas ajax al endpoint de la aplicación *user analysis notifications*. Este endpoint consulta en la base de datos por el estado del proceso, y retorna una estuctura con el porcentaje de avance y la descripción del estado.

- *analysisresults.html*: Esta vista despliega los resultados del análisis al usuario por medio de una gráfica, así como un listado de los amigos del usuario. Estos se encuentran ordenados del más similar al menos similar de acuerdo a los factores de sus personalidades con respecto a los del usuario.

5.3. Particularidades sobre consulta de embeddings

Para realizar la representación de los textos de los usuarios analizados como vectores Word2Vec, se usa la biblioteca *gensim* [8]. Esta permite cargar el archivo de embeddings, con el que se realizó el entrenamiento, y obtener la representación vectorial de una palabra. El archivo de embeddings debe estar cargado en memoria para poder hacer las consultas necesarias. Ahora bien, debido a la naturaleza de la composición de los módulos de Flask, el módulo responsable de cargar los embeddings en memoria necesita ser instanciado con cada request. En otras palabras, se requiere cargar el archivo de embeddings cada vez que un usuario desee realizar un análisis. Esto es inconveniente ya que, debido al tamaño del archivo (unos 6gb aproximadamente), se requiere de tiempo adicional para cargar los embeddings. Dependiendo de la computadora en la que se encuentre montada la aplicación, el tiempo de carga puede ir de 6 a 30 minutos. Ese tiempo es inaceptable para un proceso en un ambiente web. Por esta razón, se planteó la separación de los procesos de carga y consulta de embeddings.

Se propuso un módulo que funcione como servidor y otro módulo cliente. Ambos fueron desarrollados con sockets de Python. El proceso servidor, *EmbSocketServer*, carga el archivo de embeddings, luego abre un socket escuchando en un puerto de la dirección local de la computadora. Por su parte, el módulo cliente, *EmbSocketClient*, abre otro socket para enviar solicitudes a la máquina local al puerto en el que escucha el proceso servidor. Así, el módulo *EmbSocketClient* puede solicitar por el vector correspondiente a cada palabra del texto del usuario. Lo anterior permite que el archivo se embeddings se cargue solo una vez al iniciar el proceso servidor *EmbSocketServer*, mientras se mantiene la

[8] https://radimrehurek.com/gensim/

funcionalidad de consulta de vectores de palabras.

El módulo cliente se integró en del módulo que se ejecuta en la cola de Redis. El esquema completo de las llamadas a consultas al proceso servidor se muestra en la figura 10.

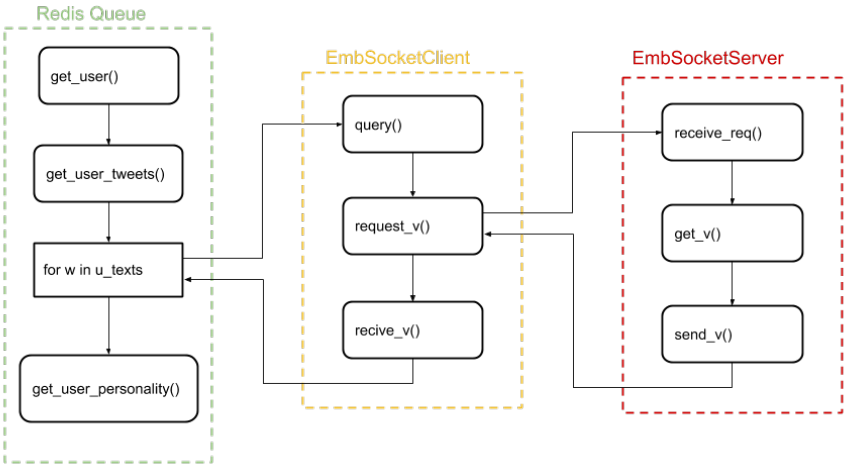

Figura 10: Estructura de consulta a servidor con los embeddings

En verde, se muestan los procesos que tienen lugar en la cola de Redis. El método *get_user* obtiene datos generales del usuario, (nombre, cantidad de tuits y número de amigos). El método *get_user_tweets* obtiene los textos de los tuits del usuario. En este punto, se llama al método *query* de una instancia de *EmbSocketClient* para obtener las representaciones de vectores de cada palabra en los textos. En esta etapa se genera una representación de promedio de vectores, como se explica en el marco teórico, para obtener un solo vector para todo el conjunto de textos del usuario. Por último, el método *get_user_personality* obtiene la personalidad del usuario cargando a memoria los modelos de clasificación y realizando la predicción de acuerdo al vector que representa su conjunto de textos.

En el recuadro amarillo se describe el proceso de consulta a de un vector de embeddings de palabra. Este proceso tiene lugar en una instancia del módulo cliente *EmbSocketClient*, dentro de la cola de Redis. Primero, el proceso dentro de la cola de Redis, que obtiene textos del usuario, llama al método *query* del módulo *EmbSocketClient*. Se invoca al método *request_v* con la palabra de la cual se desea obtener un vector. Este método hace envía la petición a través de un socket hacia el módulo servidor *EmbSocketServer*. Luego, otro método *receive_v* recibe el vector de embedding de la palabra solicitada. Por último, este

vector es retornado al proceso en la cola de redis, que realiza consultas similares por cada palabra de los textos.

En el recuadro rojo , se describe el proceso del módulo servidor *EmbSocketServer*. El método *receive_req* se encuentra en continua ejecución a la espera de solicitudes del módulo cliente. Al recibir una solicitud, se invoca al método *get_v* para obtener el vector correspondiente a la palabra solicitada. Este mismo método serializa el vector como un objeto que es enviado con el método *send_v* al módulo cliente.

5.4. Estructura de proceso de análisis

Finalmente, el esquema en detalle del proceso que lleva a cabo el análisis de la personalidad de un usuario y de sus amigos queda descrito en la figura 11.

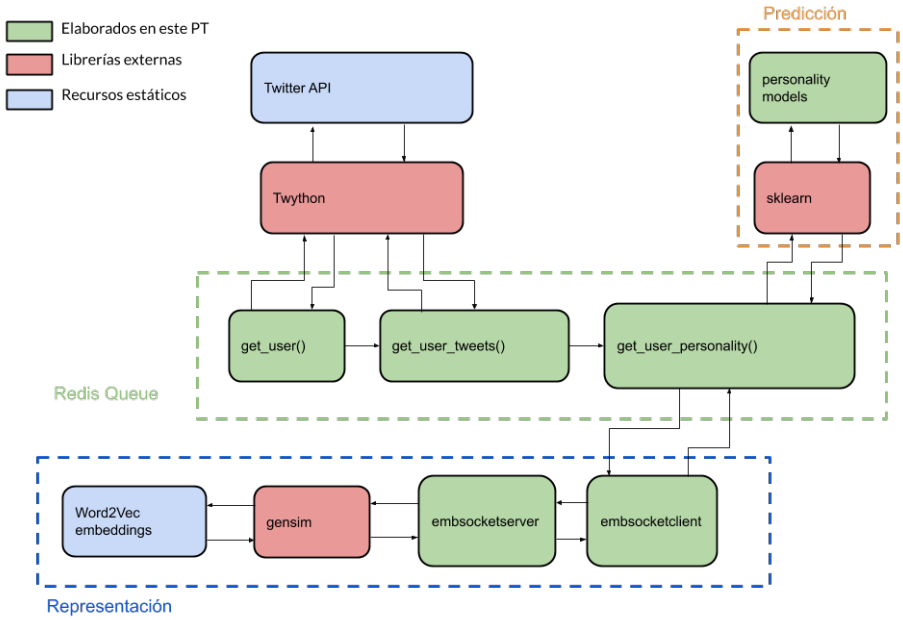

Figura 11: Detalle de estructura de proceso de análisis

El proceso inicia solicitando a la API de Twitter, por medio de la biblioteca *Twython*, los datos del usuario a analizar. También se recupera la lista de publicaciones del usuario. Luego, para obtener la personalidad del usuario se pasa a una etapa de representación. El módulo servidor tiene cargados los embeddings y los puede consultar por medio de la biblioteca *gensim*. El módulo cliente solicita los vectores correspondientes a las palabras al servidor. Una vez se tienen

todos los vectores de todas las palabras de los textos del usuario, se hace una ponderación. Un promedio de los elementos de los vectores, para producir un solo vector de trescientos elementos, es generado por usuario. Después se pasa a la etapa de generación de predicciones. En esta etapa, se utilizan los modelos entrenados para producir el vector de cinco elementos binarios, uno por cada rasgo del modelo Big Five. La tarea de análisis termina cuando tanto el usuario como sus amigos quedan representados en ese vector de rasgos de personalidad. Este vector es almacenado en la base de datos, en el modelo results, y se muestra al usuario de manera gráfica.

5.5. Capturas de la aplicación y ejemplo de uso

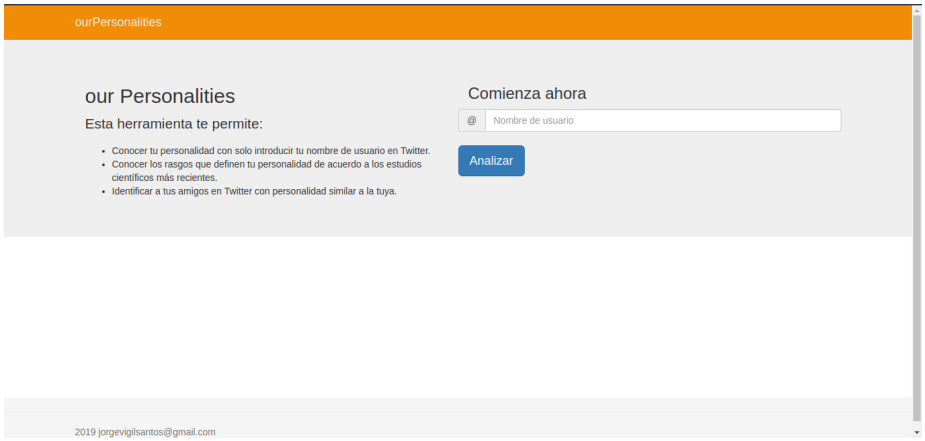

Figura 12: La página de inicio de la aplicación. Se muestra una forma para que el usuario introduzca su tag en Twitter.

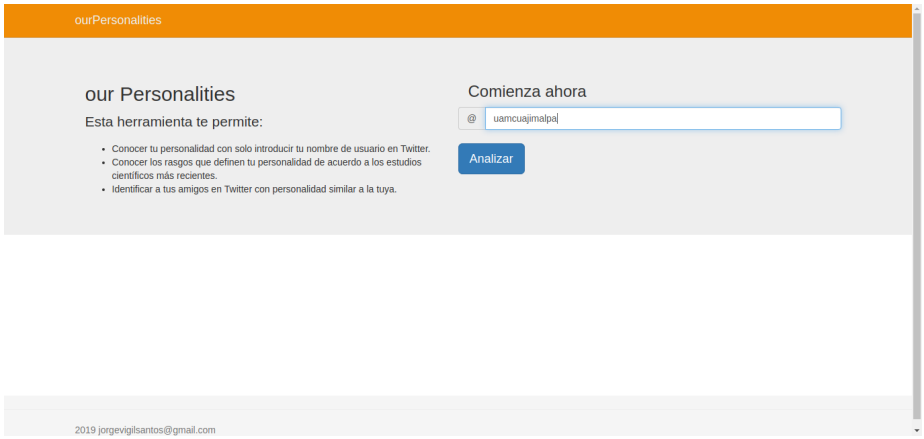

Figura 13: Como ejemplo, se introduce el nombre de usuario de la cuenta uamcuajimalpa en Twitter

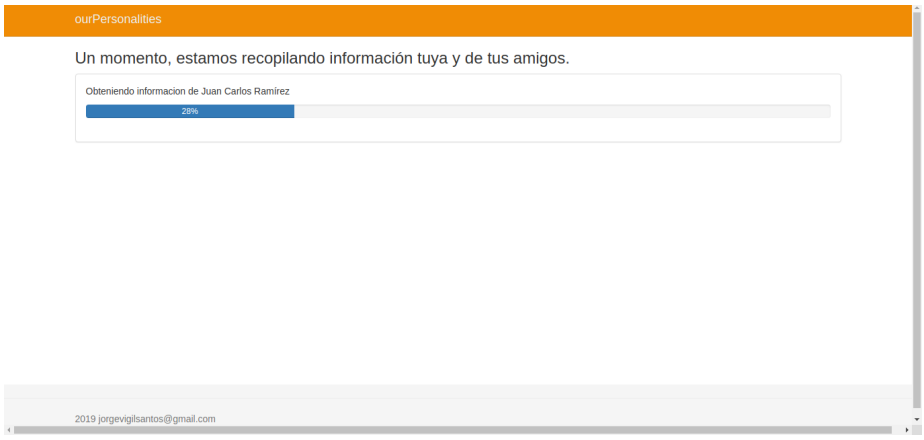

Figura 14: Progreso de análisis. Se muestra una barra de progreso que indica el porcentaje de avance del proceso.

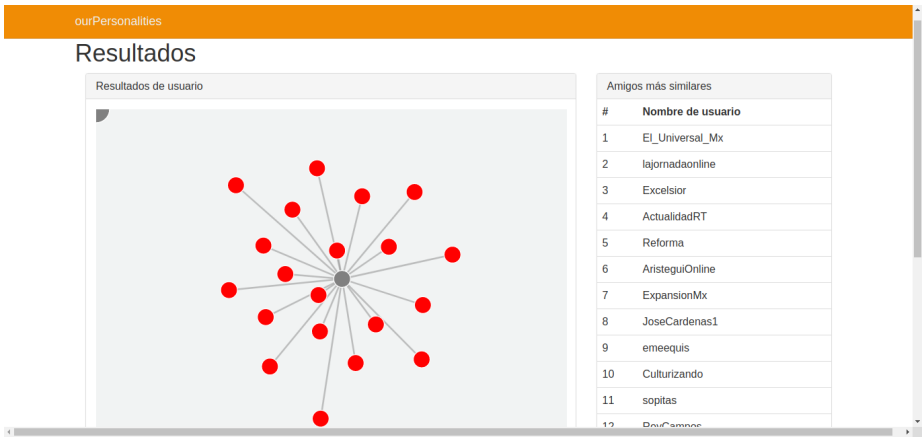

Figura 15: Vista de resultados. Se muestran los resultados del análisis por medio de una gráfica. Cada nodo en la gráfica representa un usuario. El nodo central corresponde al usuario que está siendo analizado y los rojos, corresponden a sus amigos. Los amigos más similares se muestran más cerca del usuario, mientras que los más diferentes se muestran más lejos. Del lado derecho se muestra un listado de los amigos del usuario ranqueados del más similar al menos similar respecto al perfil de personalidad.

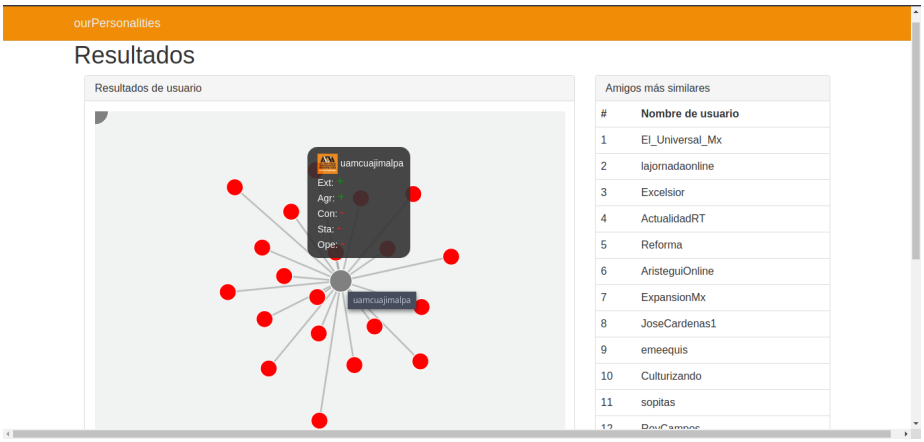

Figura 16: Un "tooltip" se muestra cuando el usuario pasa el mouse sobre los nodos. Se muestran las características de personalidad del usuario. Se representa con un '+' una polaridad positiva del rasgo indicado, y con '-' una polaridad negativa. Para que el gráfico sea más llamativo, se muestra también la foto de perfil del usuario en su cuenta de Twitter.

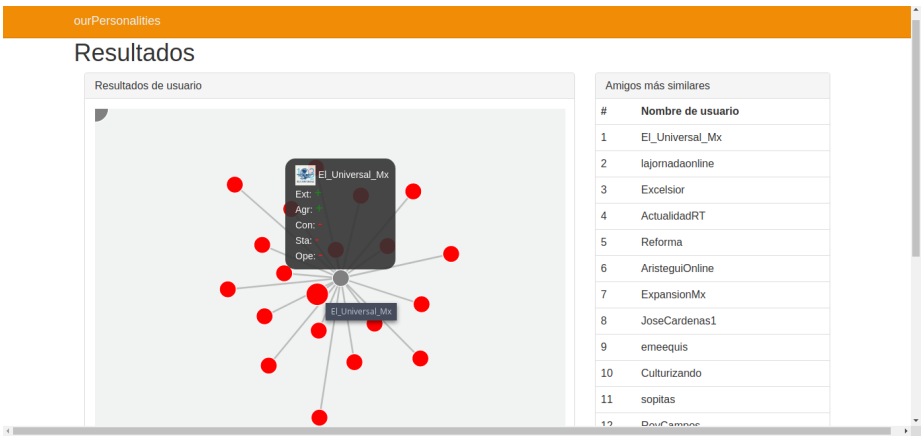

Figura 17: Otro "tooltip" es mostrado para uno de los amigos del usuario. Notar que es el amigo mas similar respecto al perfil de personalidad, esto es, cuentan con las mismas polaridades para cada rasgo y se encuentra más cerca.

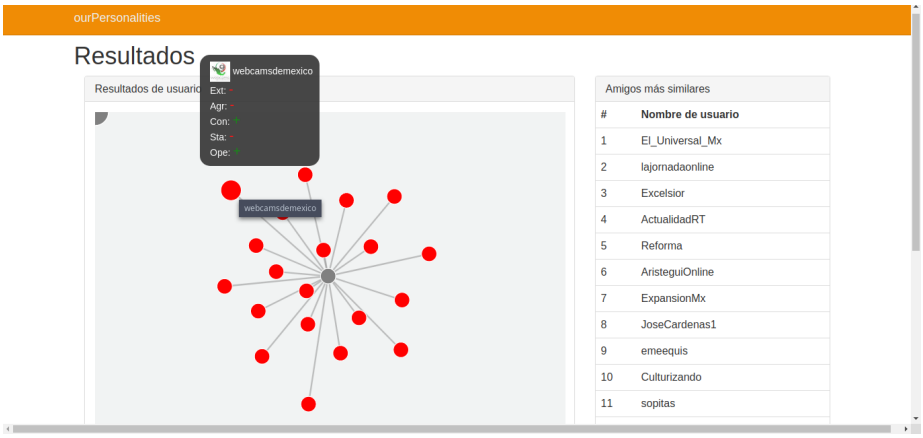

Figura 18: Otro "tooltip" mas es mostrado para otro amigo del usuario. Notar que es el amigo menos similar respecto al perfil de personalidad, esto es, las polaridades de los rasgos de personalidad son completamente opuestas a las del usuario.

6. Conclusiones

El desarrollo de esta aplicación involucró conocimientos de múltiples disciplinas, desde bases de datos, desarrollo web, hasta entrenamiento de modelos de clasificación, y por supuesto, la investigación de los elementos de psicología involucrados.

Se investigó sobre los estudios que intentan describir formalmente la personalidad por medio de rasgos. Se experimentó con modelos de aprendizaje automático para realizar clasificación de estos rasgos con base en texto. Se trabajó con diversas técnicas de representación de documentos de texto. Por último, se desarrolló una aplicación web que integra todos los elementos mencionados, y que presenta resultados cuantificables a los usuarios interesados en conocer más sobre sí mismos.

La tarea de predicción de la personalidad por medio de texto escrito es compleja. Y a pesar de que existen aproximaciones, estas han sido limitadas en resultados.

No se pretende que herramientas y trabajos así sustituyan por completo la evaluación de un especialista. Al contrario, se espera que estas herramientas den base para despertar en las personas la curiosidad sobre el conocimiento psicológico de sí mismos. De esta manera, se podría perder la estigmatización que la sociedad tiene sobre temas de psicología y trastornos mentales. Así, herramientas de este estilo podrían ayudar a una evaluación preliminar para detectar estos trastornos de manera temprana.

Como se dijo anteriormente, existe una relación entre los rasgos de la personalidad y padecimientos, tanto de naturaleza física como mental. El contar con herramientas de evaluación para detectar características que pudieran representar riesgos puede ser de ayuda a la sociedad.

Cabe destacarse la relación que existe entre relaciones de los individuos y como se ven afectadas por su personalidad. La personalidad de los individuos afecta su grado de satisfacción reportada al establecer relaciones. Sería interesante hacer uso de herramientas como la propuesta en este proyecto para estudiar la relación entre la personalidad y la calidad de la relación. Por ejemplo, determinar si ciertos factores de personalidad compartidos entre usuarios sirven como predictor de una buena satisfactoria para ambas partes.

Por último, parece importante comprender el impacto que puede tener la personalidad en cada aspecto de la vida de los individuos. Poner este conocimiento al alcance de las personas debería ser una prioridad en la investigación psicológica.

Referencias

[1] *The Cambridge Handbook of Personality Psychology*. Cambridge Handbooks in Psychology. Cambridge University Press, 2009.

[2] Mennes M DeYoung C.G Zuo X-N Kelly C-Margulies D.S Bloomfield A Gray J.R Castellanos X.F Milham M.P. Adelstein J.S, Shehzad Z. Personality is reflected in the brain's intrinsic functional architecture. *PLoS ONE*, 6:1–12, 2011.

[3] Ullman. J.D ajaraman. A. Data minig. mining of massive datasets. pages 1–17, 2011.

[4] Pierre Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, and Vibha Sinha. 25 tweets to know you: A new model to predict personality with social media. 04 2017.

[5] Preacher K Bahns A, Crandall C. Similarity in relationships as niche construction: Choice, stability, and influence within dyads in a free choice environment. 2016.

[6] Fabio Celli and Luca Polonio. Relationships between personality and interactions in facebook. pages 41–54, 06 2013.

[7] Paul T. Costa and Robert R. McCrea. *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO-FFI)*. Psychological Assessment Resources, 1992.

[8] Hugo Jair Escalante, Esaú Villatoro-tello, Antonio Juárez, Luis Villaseñor, and Manuel Montes-y Gómez. Sexual predator detection in chats with chained classifiers. pages 46–54, 2013.

[9] Janet Hernández-García, Gabriela Ramirez-de-la Rosa, Esaú Villatoro-Tello, Hector Jimenez, and Veronica Reyes-Meza. Aplicación web para identificar personalidad, género y edad de usuarios en twitter. pages 93–106, 09 2016.

[10] Klassen. A Holder. M. D. Temperament and happiness in children. *Journal of Happiness Studies*, 2009.

[11] Mark A. Hall Ian H. Witten, Eibe Frank. *Data Mining*. Morgan Kaufmann, 2011.

[12] Jokela M Laakasuo M, Rotkirck A. The company you keep: Personality and friendship characteristics. 2016.

[13] Mikolov Tomas Le Quoc. Distributed representations of sentences and documents.

[14] Tung. Joyce Y Franz. Carol Fan. Chun-Chieh Wang. Yunpeng Smeland. Olav B Schork. Andrew Holland. Dominic Kauppi. Karolina Sanyal. Nilotpal Escott-Price. Valentina Smith. Daniel J O'Donovan. Michael Stefansson. Hreinn Bjornsdottir. Gyda Thorgeirsson. Thorgeir E Stefansson. Kari McEvoy. Linda K Dale. Anders M Andreassen. Ole A Chen. Chi-Hua Lo. Min-Tzu, Hinds. David A. Genome-wide analyses for personality traits identify six genomic loci and show correlations with psychiatric disorders. *Nature Genetics*, 49:152–156, 2016.

[15] Luciana Mariñelarena-Dondena, Edgardo Ferretti, Manolis Maragoudakis, Maximiliano Sapino, and Marcelo Luis Errecalde. Predicting depression: A comparative study of machine learning approaches based on language usage. pages 42–52, 2017.

[16] Benjamin P Chapman, Brent Roberts, and Paul Duberstein. Personality and longevity: Knowns, unknowns, and implications for public health and personalized medicine. *Journal of aging research*, 2011:759170, 07 2011.

[17] J.W. Pennebaker. *The Secret Life of Pronouns: What Our Words Say About Us*. Bloomsbury USA, 2011.

[18] Socher R. Pennington, J. and C. D. Manning. Glove: Global vectors for word representation. *EMNLP*, 1532-4, 2014.

[19] Gabriela Ramirez-de-la Rosa, Esaú Villatoro-Tello, Bogdan Ionescu, Hugo Jair Escalante, Sergio Escalera, Martha Larson, Henning Müller, and Isabelle Guyon. *Overview of the Multimedia Information Processing for Personality & Social Networks Analysis Contest: ICPR 2018 International Workshops, CVAUI, IWCF, and MIPPSNA, Beijing, China, August 20-24, 2018, Revised Selected Papers*, pages 127–139. 01 2019.

[20] Gabriela Ramírez-de-la-Rosa, Esaú Villatoro-Tello, and Héctor Jiménez-Salazar. TxPI-u: A resource for personality identification of undergraduates. *Journal of Intelligent & Fuzzy Systems*, 34(5):2991–3001, May 2018.

[21] William B. Swann Samuel D. Gosling, Peter J. Rentfrow. A very brief measure of the big-five personality domains. *Journal of Research in Personality*, 37:504–528, 2003.

[22] Warren T. Norman. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. 66:574–83, 07 1963.

[23] Gregory S. Corrado Jeffrey A. Dean Tomas Mikolov, Kai Chen. Computing numeric representations of words in a high-dimensional space. 2013.

[24] Sara J. Weston, Patrick L. Hill, and Joshua J. Jackson. Personality traits predict the onset of disease. *Social Psychological and Personality Science*, 6(3):309–317, 2015.

[25] Harris K. Vazire S. Wilson, R. E. Personality and friendship satisfaction in daily life: Do everyday social interactions account for individual differences in friendship satisfaction? *European Journal of Personality*, 29:173–186, 2015.

[26] C. Wrzus and M.R. Mehl. Lab and/or field? measuring personality processes and their social consequences. *European Journal of Personality*, 29(2):250–271, 2015.

[27] Wu Youyou, David Stillwell, H. Andrew Schwartz, and Michal Kosinski. Birds of a feather do flock together: Behavior-based personality-assessment method reveals personality similarity among couples and friends. *Psychological Science*, 28(3):276–284, 2017.

[28] Whelan. D Zelenski. J, Santoro. M. Would introverts be better off if they acted more like extraverts? exploring emotional and cognitive consequences of counterdispositional behavior. *Emotion*, 2012.