Escuela Politécnica Superior de Jaén
Máster Universitario en Ingeniería Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

DETECCIÓN DEL ODIO CONTRA INMIGRANTES Y MUJERES EN COMENTARIOS DE REDES SOCIALES

Alumno/a: **Moya Pareja, Sebastián**

Tutor/a: Prof. Doña Mª Teresa Martín Valdivia
Prof. Doña Flor Miriam Plaza del Arco
Dpto.: Informática

Noviembre, 2020

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña Mª Teresa Martín Valdivia y Doña Flor Miríam Plaza del Arco, tutoras del Proyecto Fin de Carrera titulado: Detección del odio contra inmigrantes y mujeres en comentarios de redes sociales, que presenta Sebastián Moya Pareja, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Noviembre de 2020

El alumno: Los tutores:

MARTIN VALDIVIA MARIA TERESA - 26006595N

Firmado digitalmente por MARTIN VALDIVIA MARIA TERESA - 26006595N
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-26006595N, givenName=MARIA TERESA, sn=MARTIN VALDIVIA, cn=MARTIN VALDIVIA MARIA TERESA - 26006595N
Fecha: 2020.11.17 09:42:29 +01'00'

PLAZA DEL ARCO FLOR MIRIAN - 15519730C

Firmado digitalmente por PLAZA DEL ARCO FLOR MIRIAN - 15519730C
Fecha: 2020.11.17 09:19:57 +01'00'

Índice

1. INTRODUCCIÓN

1.1. Introducción

El odio es un sentimiento de antipatía, hostilidad, resentimiento, rencor, enemistad o repulsión hacia una persona. Lo que genera un gran sentimiento de enemistad y rechazo que conduce al mal hacia una persona. El odio se describe con frecuencia como lo contrario del amor o amistad.

Este sentimiento parece ser tan irracional como el amor, por lo que conduce al individuo a conductas heroicas o malvadas. La diferencia entre estas dos emociones radica en que el amor inhibe parte de las zonas donde procesan las ideas racionales y el odio las hiperactiva.

El discurso de odio es el fomento, promoción o instigación del odio, la humillación o el menosprecio de una persona o grupo de personas, así como el acoso, descrédito, difusión de estereotipos negativos, estigmatización y amenaza con respecto a dicha persona o grupo de personas y la justificación de esas manifestaciones por razones de “raza”, color, ascendencia, origen nacional o étnico, edad, discapacidad, lengua, religión o creencias, sexo género, identidad de género, orientación sexual y otras características o condición personales.

En los últimos años, se está presenciando una inquietante oleada de xenofobia, racismo e intolerancia. El odio se está generalizando, tanto en democracias liberales como en sistemas autoritarios. El problema está siendo general en toda la humanidad.

El acceso a Internet y las redes sociales están cambiando la forma de comunicación entre las personas. Dado el aumento enorme en la cantidad de contenido de los usuarios (*Ilustración 1.1*) en la web, se está generando un problema a la hora de detectar estos mensajes de ocio. Otros aspectos como la viralidad o el anonimato están generando un peligro cada vez más peligroso hacia los grupos sociales más desfavorecidos.

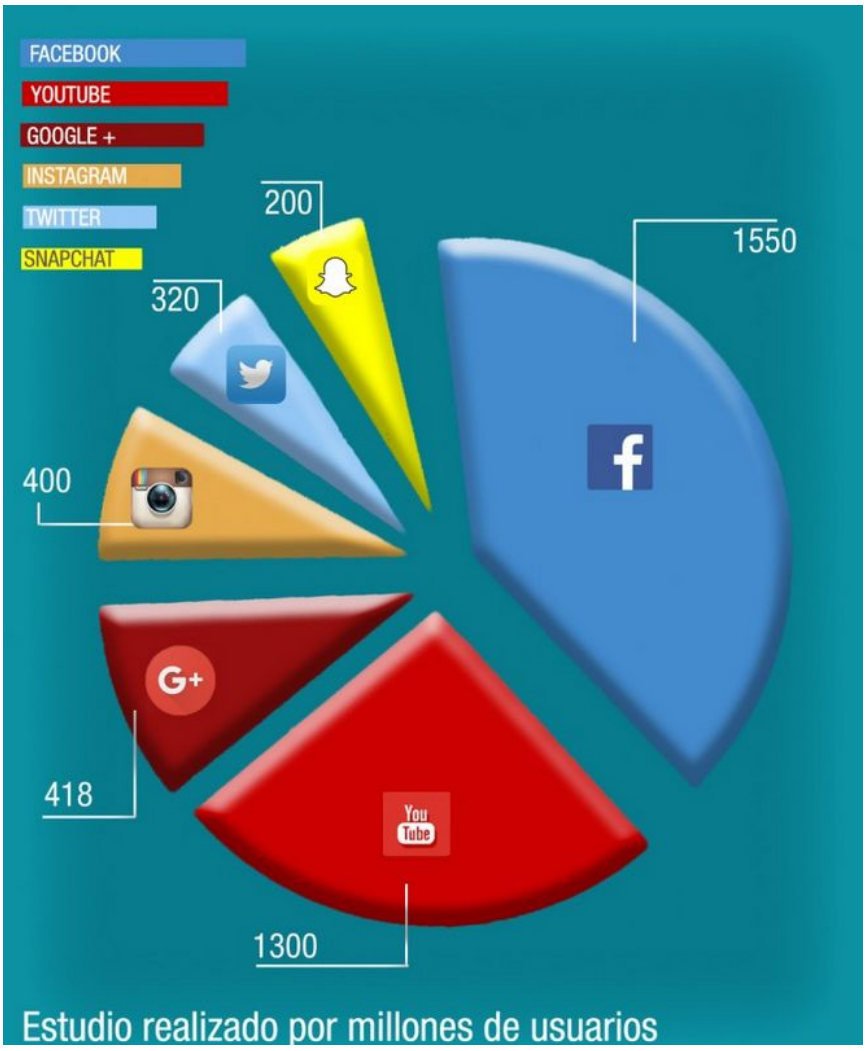

Ilustración 1.1. Número usuarios por red social en 2016 de Multiplicalia

Debido a la importancia de la detección rápida de estos mensajes de odio hacia los grupos más vulnerables, es necesario la detección de odio también conocido como *hate speech.* El *hate speech* es un tarea estudiada en el Procesamiento del Lenguaje Natural (PLN) que trata de detectar odio en un texto. Es un tarea reciente dentro del PLN por lo que todavía tiene un largo camino por recorrer.

Entre los grupos más afectados por este discurso se encuentran las mujeres y los inmigrantes. El primer grupo, es un problema conocido desde hace muchos años que está aumentando debido a la facilidad de comunicación en la actualidad.

El segundo, aumentó especialmente a partir de la crisis de los refugiados de 2015 con la llegada de casi un millón de personas que huían de la guerra civil de Siria a las que se le unieron personas que huían de conflictos en Oriente Medio y África Subsahariana.

1.2. Motivación

En los últimos años han aumentado los casos de discursos de odio o incluso los crímenes de odio para calificar discursos, mensajes o acciones que discriminan, insultan y agreden. Estos términos no deben ser entendidos como el significado genérico de la palabra “odio” que significa antipatía y aversión hacia algo o hacia alguien cuyo mal se desea[1]. Este término se centró en una connotación emotiva, pero no genérica como sufren algunas personas o grupos específicos.

Ilustración 1.2. Pirámide del odio

El discurso de odio envenena nuestro entorno a diario, ya sean en forma de pequeños gestos cotidianos o en forma de grandes campañas, especialmente

contra los colectivos más vulnerables. Este problema aumenta con mucha más fuerza en las redes sociales, ya que Internet da un anonimato de manera que pueden esconderse mientras agreden a sus víctimas. Como se puede observar en la *Ilustración 1.3*, cada año están aumentando los casos de discurso de odio detectados en Facebook.

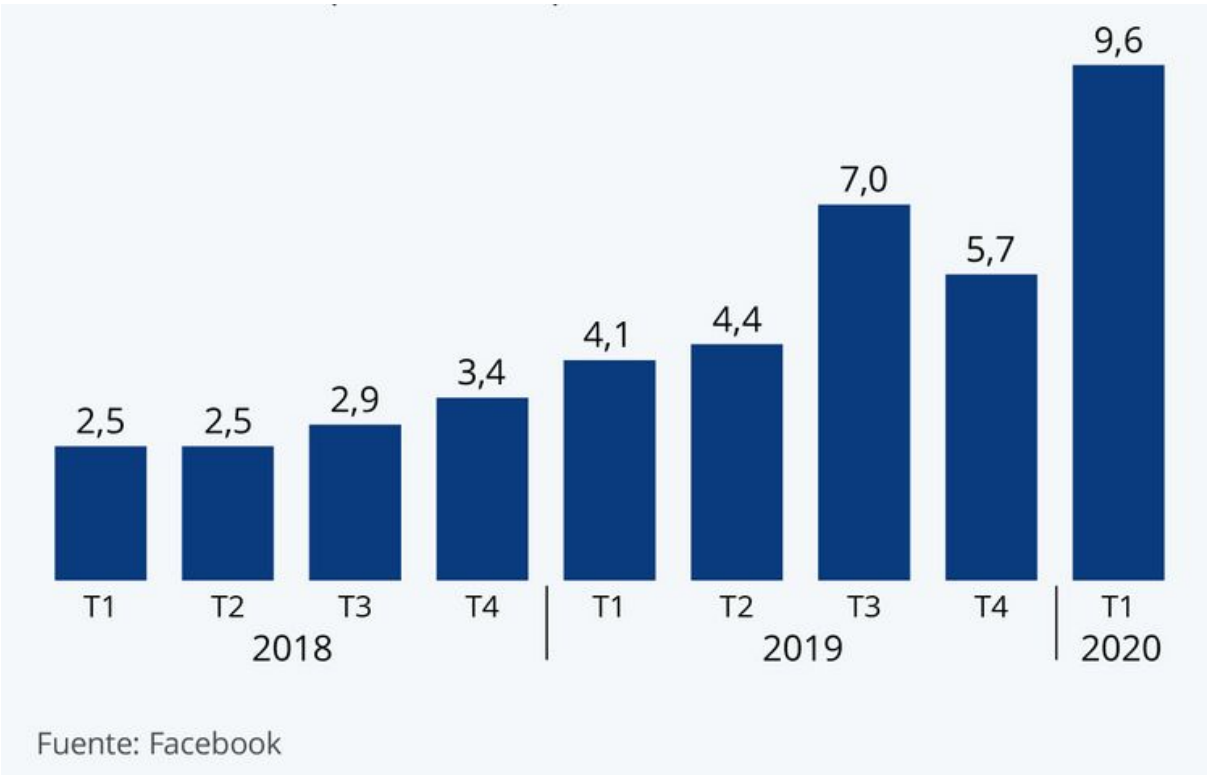

Ilustración 1.3. Nº publicaciones discurso de odio eliminadas en Facebook

La necesidad de la detección temprana del odio ha hecho que las grandes entidades tanto públicas como privadas inviertan en este campo. Uno de los ejemplos fue que la Comisión Europea hizo público un "Código de conducta" en 2016. Este documento consistía en una serie de compromisos para luchar contra la propagación del odio en Internet en la Unión Europea. Este acuerdo permitía la posibilidad de actuar judicialmente contra estas conductas de odio conforme a la legislación de cada país. Este acuerdo fue firmado por las principales redes sociales como Facebook, Twitter o YouTube.

Recientemente se han celebrado un gran número de eventos y tareas dentro de la comunidad del PLN sobre la identificación del lenguaje ofensivo como las siguientes: *Workshop on Abusive Language [1]*, *First Workshop on Trolling, Aggression and Cyberbullying [2]*, que ha incluido tareas de identificación de la agresión, *GermEval Shared Task on the Identification of Offensive Language [3]*.

El taller *Workshop on Abusive Language* fue celebrado en ACL 2017[1] en Vancouver, Canadá constaba de 14 participaciones, diez de las presentaciones eran orales y cuatro en formato póster. Las ponencias del taller abarcan una amplia gama de temas: detección del lenguaje abusivo en diferentes idiomas, análisis del lenguaje abusivo en diferentes dominios, elaboración de corpus y etiquetado, etc.

El taller *First Workshop on Trolling, Aggression and Cyberbullying* celebrado el 25 de agosto de 2018 en EEUU. La tarea consistía en desarrollar un clasificador que pudiera discriminar entre textos abiertamente agresivos, agresivos en cubierto o no agresivos. Para esta tarea, se utilizó un conjunto de datos de 15000 mensajes de Facebook etiquetados con agresión.

La tarea *GermEval Shared Task on the Identification of Offensive Language* trata de clasificar *tweets* en alemán compuesto por dos tareas: una de clasificación binaria y otra de clasificación multiclase.

Este proyecto se va a apoyar en la tarea 5 de SemEval 2019: *Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter [4].* Esta tarea trata sobre la detección de *hate speech* sobre inmigrantes y mujeres en mensajes extraídos de Twitter en español e inglés. Se ha seleccionado para trabajar el español, ya que es un idioma en el que el número de recursos de detección de odio es limitado a pesar de ser la tercera lengua más utilizada en Internet como indica la *Ilustración 1.4*.

[1] https://www.aclweb.org/portal/

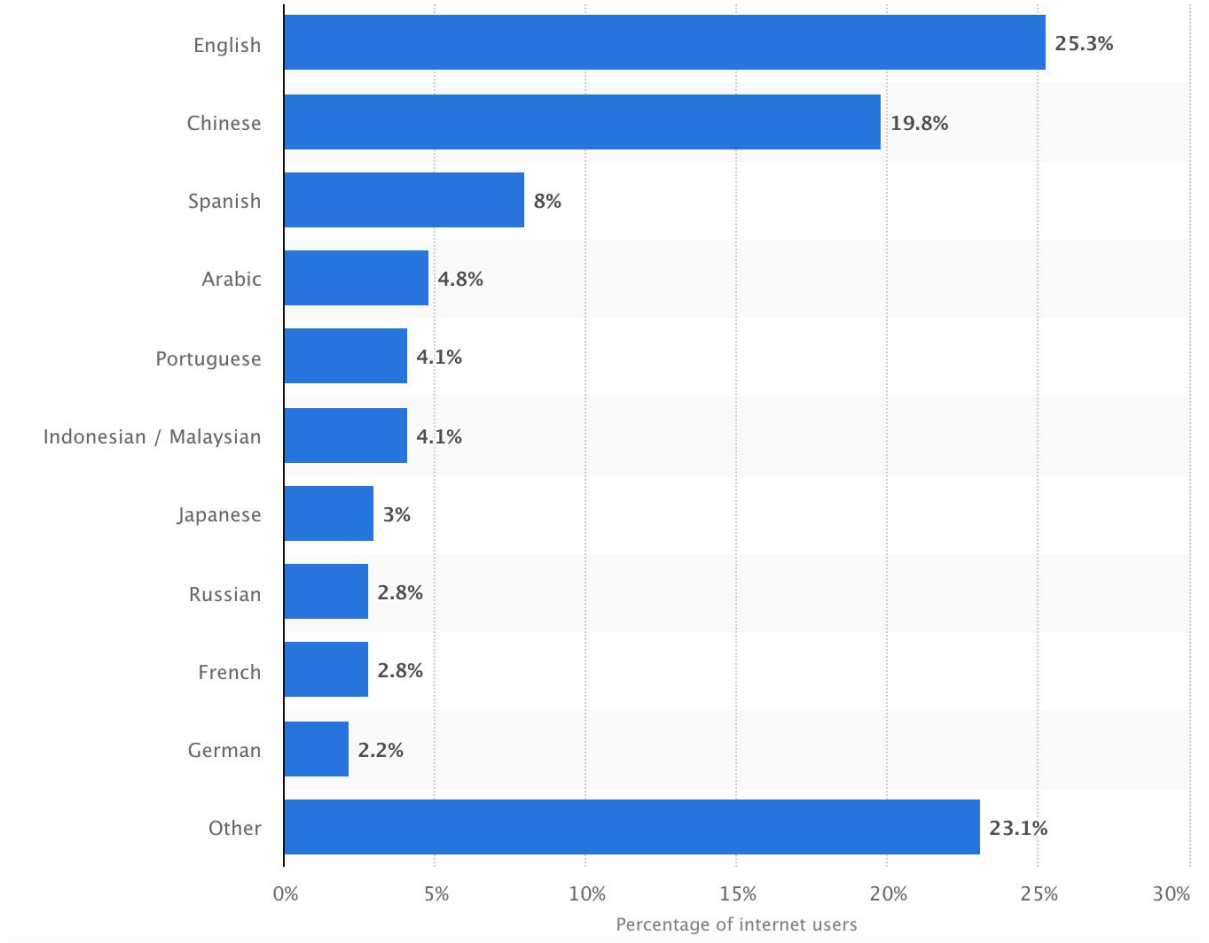

Ilustración 1.4. Lenguas más utilizadas en Internet en 2020

1.3. Propósito del proyecto

En primer lugar, será necesario estudiar los conjuntos de datos etiquetados que más se adapten al tipo de textos de la red social Twitter. En segundo lugar, habrá que realizar un estudio de los distintos recursos de detección de odio en español. Una vez analizados estos recursos, se evaluarán cuáles son las técnicas de PLN más empleadas y más efectivas para la detección de odio en las redes sociales. A continuación, se generará un sistema que combine el uso de las distintas técnicas estudiadas anteriormente.

Por último, para que el usuario pueda utilizar la herramienta de forma más sencilla y visual, se desarrollará un portal web que permita al usuario interactuar, realizando análisis de los textos que se le proporcione.

1.4. Objetivos

Los objetivos del trabajo serán:

1. Estudiar los diferentes conjuntos de datos etiquetados para el análisis de odio en Twitter.
2. Estudiar las diferentes técnicas de PLN utilizadas para la identificación del discurso de odio.
3. Tratar el texto mediante técnicas de PLN (tokenización, stemming, stopwords…).
4. Integrar diferentes técnicas de PLN para el análisis de odio creando un sistema propio.
5. Comparar los resultados del sistema desarrollado en el proyecto con los resultados obtenidos por los participantes de la tarea HatEval (Tarea 5 Semeval 2019).
6. Extraer tweets en directo mediante la API de Twitter.
7. Analizar cuentas de Twitter para la detección de usuarios tóxicos.
8. Estudiar las distintas tecnologías para realizar el portal web.
9. Desarrollar una aplicación web para que el usuario pueda realizar la interacción con el sistema.
10. Redactar una memoria que recoja todo el trabajo desarrollado.

1.5. Resultados esperados

Tras finalizar el proyecto, se deben obtener los siguientes resultados:

- Investigación de los recursos existentes para el análisis del discurso de odio.
- Desarrollo de una herramienta de análisis del discurso de odio.

- Verificación de resultados del sistema.
- Monitor de análisis del discurso de odio sobre tweets.
- Memoria del proyecto.
- Manuales de instalación y de usuario.

1.6. Planificación

El propósito de la planificación de un proyecto es simular un entorno laboral, que permita realizar estimaciones razonables sobre el coste y el tiempo del proyecto. Estas estimaciones se podrán ir reajustando a medida que se desarrolle el proyecto.

1.6.1. Diagrama de Gantt

Para llevar a cabo esta planificación se ha decidido realizar un diagrama de Gantt ya que es una herramienta muy útil al proporcionar una vista general de las tareas programadas. Esto permite a todas las partes saber que tareas tendrán que completarse y en qué fecha.

Las ventajas de trabajar con el diagrama de Gantt son:

- Claridad
- Vista general simplificada
- Datos sobre rendimiento
- Mejor gestión del tiempo
- Flexibilidad

En la *Tabla 1.1* se observa la estimación temporal del proyecto, con su Diagrama de Gantt (*Ilustración 1.5*). Para su realización se ha utilizado la herramienta *Tom's Planner*[2].

[2] https://www.tomsplanner.es/

Nombre de tarea	Duración	Comienzo	Fin
Inicio	1 día	04/05/2020	05/05/2020
Planificación	1 días	05/05/2020	04/05/2020
Estudio de sistemas	10 días	06/05/2020	19/05/2020
Extracción corpus	3 días	20/05/2020	22/05/2020
Implementación sistema	30 días	25/05/2020	03/07/2020
Comparación resultados	2 días	06/07/2020	07/07/2020
Diseño interfaz	3 días	08/07/2020	10/07/2020
Estudio recursos interfaz	5 días	13/07/2020	17/07/2020
Implementación interfaz	10 días	20/07/2020	31/08/2020
Fase de pruebas	5 días	03/08/2020	07/08/2020
Redacción memoria	30 días	10/08/2020	18/09/2020

Tabla 1.1. Planificación del proyecto

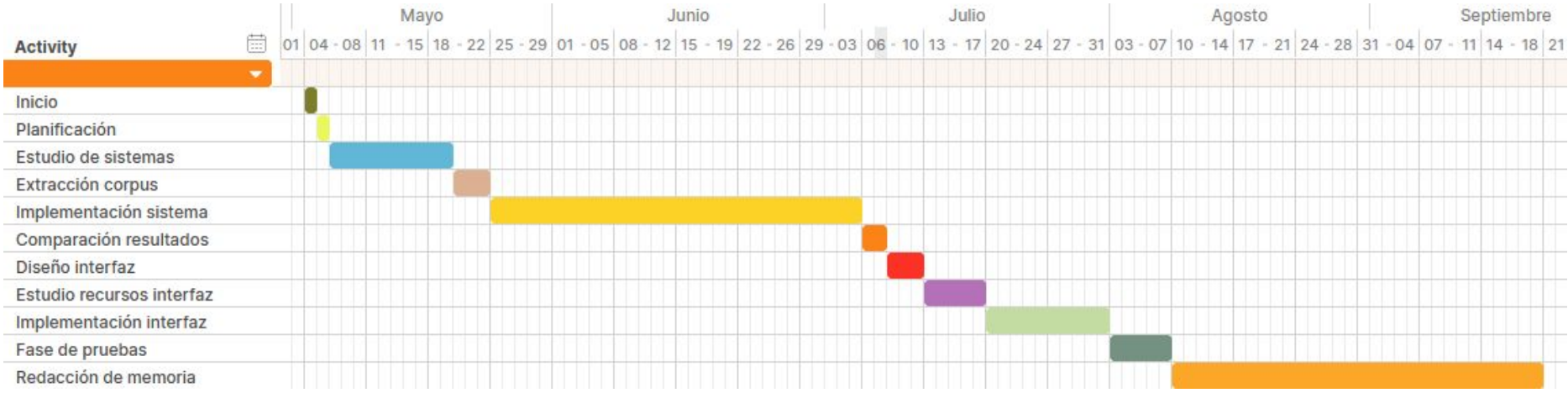

Ilustración 1.5. Planificación del proyecto

1.6.2. Estimación de costes

Cualquier proyecto software que se desarrolle implica unos costes, que se pueden dividir en costes previos, costes durante la producción y costes de mantenimiento. Dichos costes son para invertir en software, hardware, costes de personal y otros costes. Para hacer esta estimación habrá que tener en cuenta los costes mensuales y la duración del proyecto.

1.6.2.1. Costes hardware

Los costes hardware son los costes destinados para equipamiento físico del sistema informático. Las especificaciones del equipo desarrollado son las siguientes, aunque no es necesario que sean las mismas para su ejecución, ya que cualquier equipo de la actualidad es capaz de ejecutar el sistema.

- Ordenador portátil: ASUS ZenBook 14 UX431FA-AM132T …….... 749 €
    - Intel Core i5-10210U
    - 8GB LPDDR3 2133MHz
    - 512GB SSD M.2 PCIE gen3 x2 NVME
    - Tarjeta gráfica integrada Intel UHD
    - Pantalla de 14" FullHD (1920x1080)
- Monitor: Samsung S24F350 …… 149 €
    - Brillo pantalla: 250 cd/m
    - Tiempo de respuesta de 4 ms
    - Conectividad VGA y HDMI
- Periféricos: Combo Logitech MK270 …….. 34.99 €
    - Ratón
    - Teclado

Debe amortizarse durante el transcurso del proyecto. El coste total del equipo es de 932.99€. Se estima una vida útil de cuatro años. Por lo que el coste anual será de 233.25€. El proyecto tiene una fecha estimada de siete meses, el coste total del equipo hardware en el proyecto será **136.06€**.

1.6.2.2. Costes software

Los costes software son los destinados a licencias de aplicaciones informáticas para su desarrollo. Los costes de software son los siguientes.

- Windows 10 …… 44.99 €
- Paquete Microsoft Office 2018 …… 19.95€
- Microsoft Visual Code …….. 0 €
- Google Drive …….. 0 €
- Google Collab ……..... 0 €
- GibHub ……. 0 €
- Licencia de desarrollador Twitter ……. 0 €
- Lexicón misoginia ……. 0 €
- Python …… 0 €
- Acceso papers HateEval ……..... 0 €
- Mozilla Firefox …….. 0 €

El coste total de las licencias es de 69.94€. Se le estima una vida útil de tres años. El proyecto se estima que durará siete meses, por lo que el precio total del proyecto será de **12.63€**.

1.6.2.3. Costes de personal

Los costes de personal son los derivados de los salarios al personal asignados al proyecto. Aunque el proyecto va a ser realizado sólo por una persona,

que tendrá asignado varios roles. Irá variando el rol dependiendo de la tarea a realizar.

- **Analista programador:** es el encargado de realizar labores de levantamiento y análisis de requerimientos, desarrollo de software, aplicaciones y soluciones tecnológicas. Es el encargado de diseñar el sistema y obtener los algoritmos. Además debe asegurar la calidad del software y el servicio de mantenimiento.
- **Programador informático:** encargado de escribir, probar y almacenar los registros para implementar el sistema diseñado previamente.

Para definir el salario de cada empleado, se ha recurrido al *Convenio Colectivo de Sector de OFICINAS Y DESPACHOS (23000245011981) de Jaén* a partir de enero de 2019. Se establecen los siguientes salarios mínimos anuales:

Puesto	**Sueldo mensual**	**Sueldo hora**
Analista	1517.19€	9.48€
Programador	1430.49€	8.94€

Tabla 1.2. Sueldos Convenio Oficinas y Despachos de Jaén

La asignación de tiempo de cada tarea se muestra en la *Tabla 1.3*.

Nombre tarea	**Duración**	**Rol**
Definición problema	2 días	Analista
Estudio de sistemas	10 días	Analista
Estudio recursos aplicación web	5 días	Analista
Obtención y preparación Corpus	3 días	Programador
Desarrollo sistema *hate speech*	30 días	Programador
Comparación resultados sistema y pruebas	4 días	Analista

Diseño aplicación web	3 días	Analista
Implementación aplicación web	10 días	Programador
Pruebas aplicación web	1 días	Analista
Manual de usuario	3 días	Analista
Manual de instalación	2 días	Analista
Redacción documentación	25 días	Analista

Tabla 1.3. Tareas, duración y rol asignado del proyecto

En la *Tabla 1.4* se muestra el resumen del total de horas y el coste por cada miembro.

Personal	**Total de horas**	**Coste Total**
Analista	440	4171.20€
Programador	344	3075.36€
Total	584	72460.56€

Tabla 1.4. Costes de personal

El coste total estimado en persona es de **7246.56€** distribuido en 784 horas.

1.6.2.4. Otros costes

Otros costes derivados del proyecto que no han sido contemplados ni como costes hardware ni software, es la conexión a Internet. Una conexión de 300Mb en la compañía Movistar tiene un coste de 38€/mes. El coste total para el proyecto será de **266€**.

1.6.2.5. Coste total

El coste total del proyecto es el que engloba los costes hardware, los costes software, los costes de personal y otros costes. Sobre el coste del desarrollo del

proyecto se añade un beneficio del 10%. En la *Tabla 1.5.* se puede observar con detalle.

Concepto	**Coste**
Costes hardware	136.06€
Costes software	12.63€
Costes de personal	7246.56 €
Otros costes	266€
Coste del proyecto	7661.25€
Beneficio del 10%	766.13€
Coste total	8427.38€

Tabla 1.5. Coste total del proyecto

El coste total del proyecto es de **8427.38€**.

2. DISCURSO DE ODIO

2.1. ¿Qué es el discurso de odio?

A continuación, se van a presentar diferentes definiciones existentes sobre la detección de odio, también conocida como *hate speech* en inglés.

La *Real Academia Española*[3] en su *diccionario panhispánico del español jurídico* define el *delito de provocación al odio* como:

"Delito que cometen quienes públicamente fomentan, promueven o incitan directa o indirectamente al odio, hostilidad, discriminación o violencia contra un grupo, una parte del mismo o una persona determinada por razón de su pertenencia a aquel, por motivos racistas, antisemitas u otros referentes a la ideología, religión o creencias, situación familiar, la pertenencia de sus miembros a una etnia, raza o nación, su orígen nacional, su sexo, orientación o identidad sexual, por razones de género, enfermedad o discapacidad."

El *Ministerio del Interior*[4] define el delito de odio, con la siguiente definición:

"Cualquier infracción penal, incluyendo infracciones contra las personas o las propiedades, donde la víctima, el local o el objetivo de la infracción se elija por su, real o percibida, conexión, simpatía, filiación, apoyo o pertenencia a un grupo con una característica común de sus miembros, como su raza real o perceptiva, el origen nacional o étnico, el lenguaje, el color, la religión, el sexo, la edad, la discapacidad intelectual o física, la orientación sexual u otro factor similar."

El discurso de odio según las Naciones Unidas[5] es cualquier forma de comunicación de palabra, por escrito o a través del comportamiento, que sea un ataque o utilice un lenguaje peyorativo o discriminatorio en relación con una persona

[3] https://www.rae.es/
[4] http://www.interior.gob.es/web/servicios-al-ciudadano/delitos-de-odio/que-es-un-delito-de-odio
[5]https://www.un.org/en/genocideprevention/documents/advising-and-mobilizing/Action_plan_on_hate_speech_ES.pdf

o un grupo sobre la base de quiénes son o, en otras palabras, en razón de su religión, origen étnico, nacionalidad, raza, color, ascendencia, género u otro factor de identidad.

John T. Nockleby expone en la *Encyclopaedia of the American Constitution* otra definición diferente:

"Discurso de odio se considera usualmente a aquel que incluye expresiones de animosidad o menosprecio hacia un individuo o grupo basada en una característica compartida por un grupo como la raza, color de piel, origen nacional o étnico, género, invalidez, religión,u orientación sexual."

Las cuatro definiciones conceptualmente son similares. Todas ellas hacen referencia a actitudes violentas, de amenaza u hostil. También hacen alusión a que el discurso de odio puede estar dirigido tanto a un individuo como a un grupo de personas que tengan en común la forma de pensar, su origen o su físico. La diferencia son las características elegidas para que un discurso se considere de odio o no. Esto puede ser debido a que el concepto de odio es tan genérico y amplio que es difícil concretar una idea respecto al significado de esa palabra.

Como hemos visto no hay una definición exacta para el discurso de odio, por ello se ha decidido acordar un concepto para el discurso de odio durante este estudio de carácter científico.

La definición de *hate speech* tomada para este trabajo es la definida en el taller *SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter.* Con el fín de tener un significado común para los participantes, la competición decidió poner unos límites para tener un punto común entre los participantes. Para que un texto sea considerado discurso de odio debe cumplir:

- El mensaje debe tener como objetivo principal el grupo al que se dirige el discurso de odio.
- El mensaje debe de promover o incitar el odio hacia el objetivo; o humillar, despreciar o amenazar al objetivo.

Con el objetivo de dejar más claro el concepto seleccionado como detección de odio, como puede ser el lenguaje ofensivo o blasfemar, se van a mostrar unos ejemplos.

Antes de seguir, informar que en la siguiente sección va a mostrar contenido bastante agresivo por lo que puede ser sensible para el espectador. Si lo desea, puede seguir en la *sección 2.3*.

En el *tweet* de la *Ilustración 2.1* podemos observar un ejemplo de discurso de odio sin utilizar lenguaje ofensivo. Se observa cómo habla a una persona del sexo femenino de una forma despectiva intentando humillarla. La clave del mensaje ofensivo es una expresión muy utilizada en contra de la mujer como es “vete a lavar platos y limpiar la casa”.

Ilustración 2.1. Ejemplo tweet hate speech

En la *Ilustración 2.2* se puede observar un mensaje sin discurso de odio, pero con lenguaje ofensivo. Aparentemente podría parecer que sí hay *hate speech.* Aunque ataca a una persona en concreto, como es una mujer, se puede observar que no la ataca por pertenecer a un grupo social vulnerable como sería en este caso la mujer. Las palabras “maldita”, “hipócrita” o “falsa” no son insultos que dependan del género o del grupo social, sino que se utilizan en cualquier ocasión. Por lo tanto,

aunque sea un mensaje de odio hacia una mujer, se puede observar que no es colectivo, por lo que no se considera como un discurso de odio hacia la mujer.

Ilustración 2.2. Ejemplo tweet no hate speech

En el *tweet* de la *Ilustración 2.3* se puede observar un mensaje que no contiene discurso de odio aunque contiene lenguaje ofensivo. La expresión “la concha de tu madre” es un insulto, aunque se puede observar que no va a nadie en concreto. Si pudiese ir a alguien en concreto, tampoco se observa que vaya a un grupo determinado para que se considere *hate speech*.

Ilustración 2.3. Ejemplo ilustración lenguaje ofensivo

Como se ha podido apreciar, la detección de odio es una tarea muy compleja y subjetiva. Es una tarea difícil de identificar debido a que cada persona dependiendo de la educación y nivel social que tenga, podrá considerar o no como *hate speech* el mismo mensaje. Es más, dependiendo del día y de la situación, la misma persona podrá interpretar de diferente manera el mismo mensaje.

En base al trabajo presentado, se ha decidido acortar que se va a considerar discurso de odio o *hate speech.* Se va a establecer como discurso de odio cualquier mensaje que haya antipatía, desprecio, repulsión o discriminación contra un individuo o grupo de personas protegidas como mujeres o inmigrantes.

2.2. Recursos

Uno de los pilares fundamentales en la investigación relacionada con el PLN y concretamente en la detección del discurso de odio se centra en los recursos lingüísticos. A continuación, se van a describir algunos de los recursos existentes para diferentes idiomas.

2.2.1. Hatebase

Hatebase es un proyecto de *Sentinel Project* que se describe como una base de datos de discurso de odio, estructurada y multilingüe. Hatebase utiliza el análisis del texto del discurso y la identificación de patrones de discurso de odio para poder predecir la violencia. Sus algoritmos analizan las conversaciones públicas utilizando un amplio vocabulario basado en la nacionalidad, etnia, religión, género, orientación sexual, discapacidad y clase, con datos en más de 95 idiomas y más de 175 países.

Hatebase dispone de una API para interactuar con la herramienta, que se divide en dos pasos. El primer paso es el encargado de la autenticación, en la que recibe la clave API del usuario y devuelve un *token* que dura alrededor de una hora. El segundo paso es la propia consulta, en la que recibe la consulta y el *token* de consulta y devuelve el resultado de la consulta como se puede observar en la *Ilustración 2.4.*

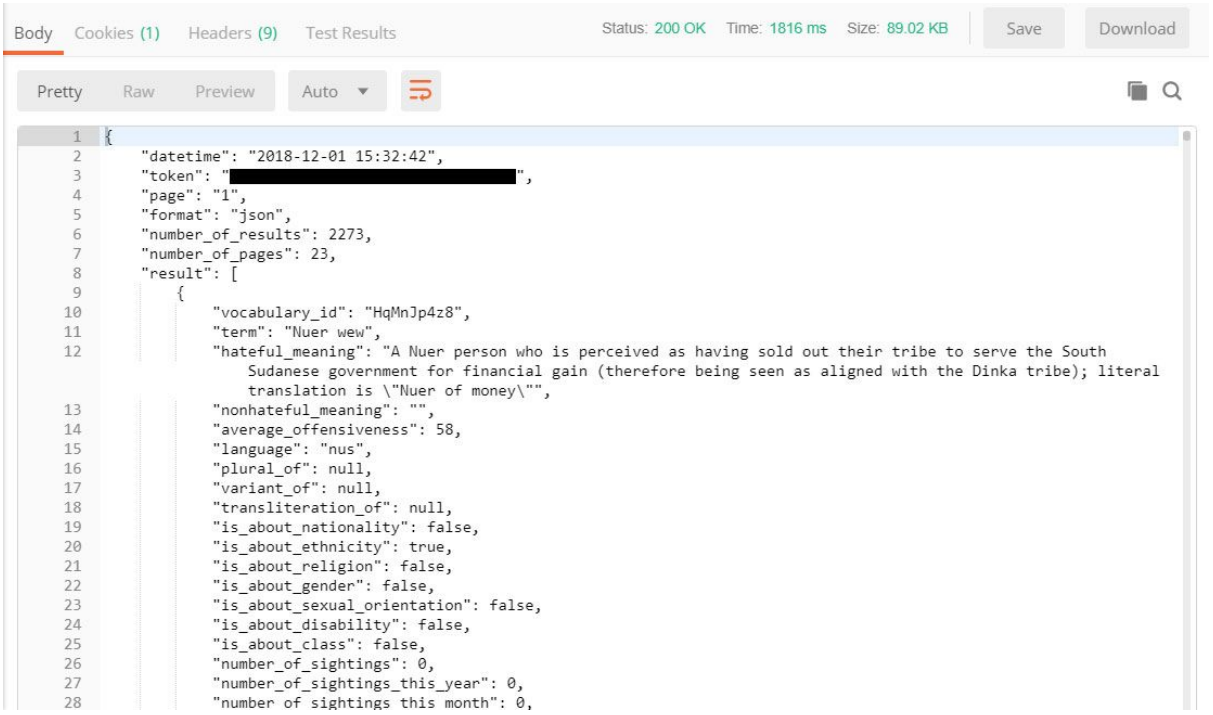

Ilustración 2.4. Ejemplo consulta API Hatebase

2.2.2. Lexicón en español de discurso de odio

Flor-Miriam Plaza et Al. [5] han desarrollado un lexicón[6] de palabras en español utilizadas en Twitter para expresar discurso de odio. Se ha analizado el discurso de tweets contra las mujeres e inmigrantes realizando experimentos de clasificación con diferentes enfoques creando un recurso lingüístico apropiado para la detección de odio en español. Se han desarrollado cuatro lexicones que afectan a diferentes temas:

- Lexicón de xenofobia: hace referencia al odio hacia los extranjeros. Está compuesto por 44 palabras.
- Lexicón de inmigrantes: contiene palabras que se refieren a la nacionalidad de los inmigrantes. Está compuesto por 250 palabras.
- Lexicón de misoginia: contiene palabras de odio hacia mujeres. Está compuesto por 183 palabras.

[6] https://github.com/fmplaza/hate-speech-spanish-lexicons

- Lexicón de insultos: contiene insultos en general. Está compuesto por 279 palabras.

Ilustración 2.5. Lexicón de xenofobia de Flor-Miriam Plaza et Al. [5]

2.2.3. Perspective API

Perspective API[7] ha sido desarrollada por Google haciendo uso de un motor de inteligencia artificial para evitar que comentarios ofensivos y *trolls* infantilicen el diálogo y baje la calidad de las conversaciones. Esta herramienta da diferentes puntuaciones sobre un texto, para ayudar a los moderadores a revisar los comentarios con mayor facilidad como muestra la *Ilustración 2.6.*

[7] https://www.perspectiveapi.com

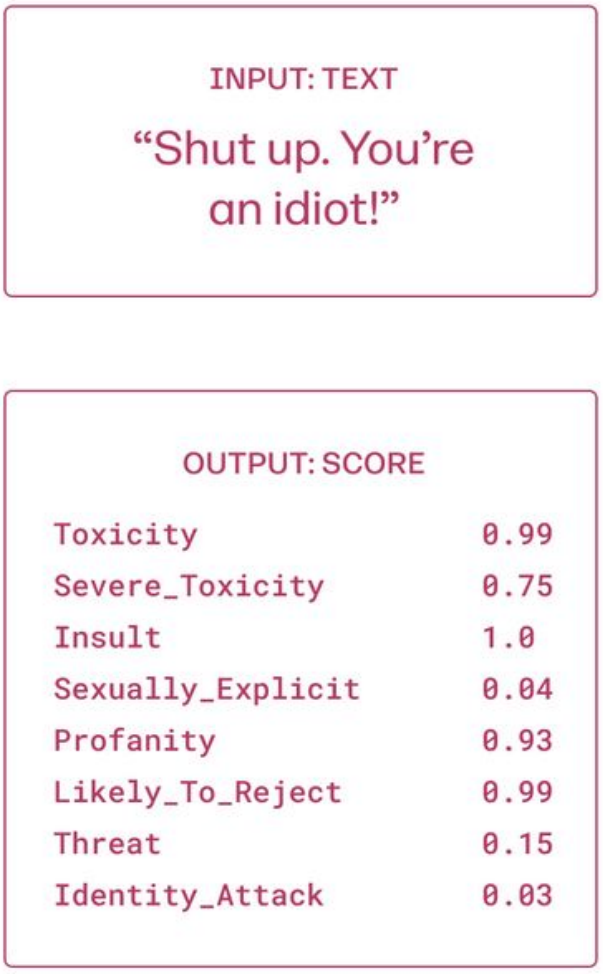

Ilustración 2.6. Ejemplo funcionamiento Perspective API

2.3. Trabajos previos

Una vez se ha entendido que es la detección de *hate speech* y utilizando los conceptos que se utilizan en el ámbito del PLN y el aprendizaje automático, se va a comenzar a realizar un estudio sobre el estado del arte en la detección del discurso de odio.

Para la detección de los mensajes con discurso de odio, se han utilizado diferentes técnicas dentro del Procesamiento del Lenguaje Natural. Como suele ocurrir dentro de las tareas de clasificación, se divide en tres grandes partes: el preprocesamiento, la extracción de características (procesamiento) y el modelo de clasificación (modelo). A continuación, se van a detallar los diferentes pasos

mediante ejemplos de sistemas de clasificación de detección de discurso de odio que han competido en la tarea HatEval 2019 explicada anteriormente.

2.3.1. Atalaya

El preprocesamiento es una parte crucial de las tareas de Procesamiento de Lenguaje Natural. El sistema Atalaya [6] ha dividido este preprocesamiento en dos niveles basándose en lo desarrollado por Luque y Pérez: preprocesamiento básico y preprocesamiento orientado a análisis de sentimientos. Usarán uno u otro dependiendo de las diferentes configuraciones.

El preprocesamiento básico incluye *tokenización*, eliminación de URLs y emails y eliminación de letras repetidas en una palabra.

El sistema de preprocesamiento orientado a análisis de sentimientos consiste en convertir el texto a minúsculas, eliminación de signos de puntuación, palabras vacías y números, lematización y manejo de la negación. Para el tratamiento de la negación, se ha buscado la palabra negada y se le ha añadido el token “NOT”.

En el procesamiento se han utilizado diferentes técnicas, se seleccionarán unas u otras según la configuración. Las técnicas utilizadas en el sistema de *Atalaya* son:

- Bolsa de palabras y caracteres.
- *Word* embeddings.
- *Tweet embeddings*.
- *Embedding* dependiendo del contexto.

El enfoque más simple para representar los *tweets* ha sido la bolsa de palabras. La bolsa de palabras construye vectores para cada *token* encontrado en los datos de entrenamiento. Para un *tweet* en particular, la bolsa de palabras contiene el número de veces que ha aparecido el *token*, lo que produce vectores pequeños con una gran dimensionalidad. Los sistemas de bolsa de palabras

modificados no incluyen solamente los *token* sino que utilizan n-gramas, contadores binarios o limitación número de *tokens*.

El uso de caracteres en *tweets* puede contener información para el análisis de sentimientos, como son la presencia de caracteres repetidos en una palabra, las mayúsculas en una palabra, emoticones y signos de puntuación. Estas marcas pueden contener información de sentimientos o mostrar contenido formal, por lo que no habría sentimientos. Para detectar estas marcas se utiliza las bolsas de caracteres en sustitución de las bolsas de palabras.

Para los *word embeddings* han usado *fastText*[8], una librería que obtiene el contexto independientemente de la representación de la palabra. En vez de usar los vectores pre entrenados, han entrenado su propio sistema con noventa mil *tweets* en países hispanohablantes. Han preparado dos versiones de *datasets*: uno con el preprocesamiento básico y otro con el preprocesamiento orientado a análisis de sentimientos. Estos dos *datasets* han sido entrenados con diferentes configuraciones.

Para los *tweets embeddings* se han usado combinaciones lineales para calcular la representación de cada *tweet*. Han seguido dos enfoques simples: promedio simple y promedio ponderado.

Seguidamente se van a describir los dos sistemas utilizados en *Atalaya* para el *HatEval 2019*. Los dos modelos utilizados son: clasificadores lineales y modelos neuronales.

Los clasificadores lineales se han entrenado con los modelos implementados en scikit-learn[9]. Se ha empezado el estudio con la configuración óptima de Luque y Pérez, que combina bolsa de palabras (BoW), bolsa de caracteres (BoC) y *tweet embeddings* con la siguiente configuración:

[8] https://fasttext.cc/
[9] https://scikit-learn.org/

- Bolsa de palabras: todos los unigramas y bigramas de las palabras con cuenta binaria y TD-IDF recalculado. Para el conjunto de datos en español, proporciona 53504 características.
- Bolsa de caracteres: todos los caracteres n-gramas menores de 5 caracteres, con cuenta binaria y TD-IDF recalculado. El *dataset* español entrenado proporciona 226156 características.
- *Tweet embeddings*: procesado con *fastText* orientado a sentimientos con vectores de palabras de 50 dimensiones.

Para los modelos neuronales, se han entrenado redes neuronales recurrentes (RNNs) utilizando representaciones dependientes del contexto en español. El primer modelo considerado fue un modelo bidireccional
LSTM con una capa densa en la parte superior, haciendo uso de vectores ELMo. Además, se ha probado otro sistema que consiste en una bolsa de palabras como se indica en la *Ilustración 2.7,* a este modelo se le ha llamado *LSTM-ELMo+BoW.* La capa *biLSTM* consiste en 256 unidades. La bolsa de palabras tiene 3500 n-gramas, los más frecuentes menores de 0.65, seguido por una capa con 512 unidades. Las dos últimas capas de densidad tienen 64 neuronas.

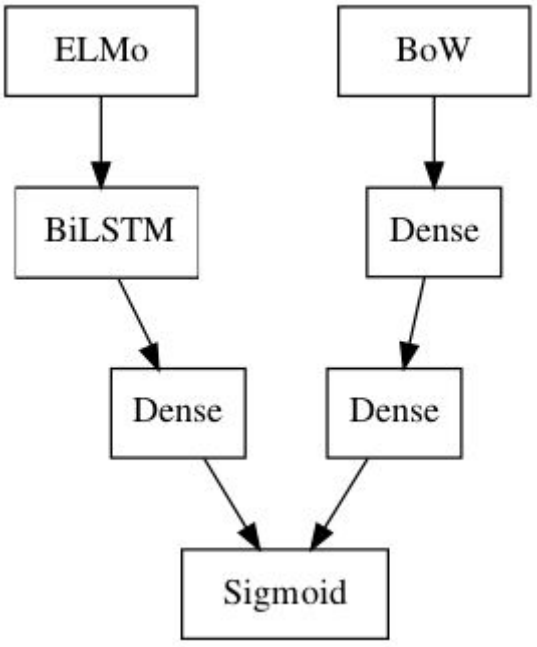

Ilustración 2.7. LSTM-ELMo+BoW arquitectura.

2.3.2. MineriaUNAM

Muchos investigadores han mostrado que el preprocesamiento es esencial para el Procesamiento del Lenguaje Natural, especialmente cuando el corpus viene del contenido de una red social. Antes de la extracción de características, MineriaUNAM [7] sigue las siguientes técnicas de preprocesamiento son aplicados para mejorar la representación de n-gramas y reducir los errores del etiquetado POS:

- Estandarización de textos a minúscula.
- Eliminación de menciones de usuario.
- Eliminación de urls.
- Reemplazo de abreviaturas a palabras reales.
- Eliminación de palabras vacías.
- Eliminación de número que un dígito (0).
- Eliminación de *hashtags* que no aparecen palabras de odio. Si aparece palabra de odio en el *hashtag* se sustituye por esa palabra.
- Sustitución de caracteres raros por espacio en blanco.
- Eliminación signos de puntuación.
- Sustitución de espacio en blanco seguidos, tabulaciones y saltos de línea por un solo espacio en blanco.

Posterior al preprocesamiento, se han extraído las siguientes características del sistema:

- **Caracteres n-gramas**: son capaces de detectar la composición morfológica. En las tareas de procesamiento de lenguaje natural existen palabras que están mal escritas, los n-gramas son muy potentes para detectar estos patrones. En este caso se va a utilizar una *n* entre 3 y 5 caracteres.

- **Palabras n-gramas**: captura la identidad de las palabras y sus vecinos. En los experimentos se ha utilizado una combinación de n-gramas que varía entre 1 y 4 palabras.
- **Pos tag n-gramas:** son secuencias de las etiquetas proporcionadas por el etiquetado gramatical. Han experimentado en varias combinaciones y se ha llegado a la conclusión que entre 2 y 4 gramas da los mejores resultados para el *dataset* utilizado.
- **N-gramas para palabras agresivas:** se ha unido el lexicón de palabras agresivas de Gomez Adorno et Al. [8] y se le han incorporado palabras extraídas del corpus de entrenamiento. Se han construido bigramas y trigramas solo con las palabras que aparecen en este lexicón.
- **N-gramas de símbolos de puntuación:** con esta característica se acerca a la coherencia y cohesión del texto. Ha ayudado a detectar ciertos patrones en el análisis del odio y agresividad. Con el corpus que se ha trabajado el mejor resultado ha sido con n-gramas de 2 a 4.

Más tarde, se ha utilizado un *framework* desarrollado por Tellez et Al. [9]. Este *framework* aproxima cualquier clasificación de texto como un problema de optimización continua. Una vez se seleccionan las características se utiliza un SM lineal.

2.4. Competiciones

2.4.1. HateSpeeDe

HaSpeeDe[10] (*Hate Speech Detection,* según el nombre en inglés) fue una propuesta de Evalita 2018[11] . El objetivo de esta tarea es detectar el discurso de odio en italiano en redes sociales: Twitter y Facebook.

[10] http://www.di.unito.it/~tutreeb/haspeede-evalita20/index.html
[11] http://www.evalita.it/2018

La tarea se subdivide en tres subtareas, determinado por las diferentes fuentes de entrenamiento utilizada para los sistemas.

- HaSpeeDe-FB: se utilizan posts de Facebook como fuente de entrenamiento y evaluación de los sistemas.
- HaSpeeDe-TW: se utilizan *tweets* como fuente de entrenamiento y evaluación de los sistemas.
- Croos-HaSpeeDe: el objetivo de esta subtarea es evaluar el comportamiento de los sistemas utilizando una fuente de entrenamiento y otra de evaluación.
    - Croos-HaSpeeDe-FB: se utilizan los textos de Facebook como fuente de entrenamiento mientras que los textos de Twitter son utilizados para la evaluación.
    - Croos-HaSpeeDe-TW: se utilizan los textos de Twitter como fuente de entrenamiento mientras que los textos de Facebook son utilizados para la evaluación.

2.4.2. MEX-A3T

MEX-A3T [15] (*Authorship and aggressiveness analysis in Mexican Spanish Tweets* en inglés) fue una tarea propuesta en IberEval 2018[12]. El objetivo de esta tarea se enfoca en el análisis de textos escritos, concretamente en castellano en el dialecto mexicano.

La tarea está subdividida en las siguientes tareas, determinando diferentes puntos de atención:

- Perfil de autor: el propósito de la subtarea es caracterizar a los usuarios identificando su lugar de residencia y su ocupación.
- Detección de agresividad: el objetivo de esta subtarea es discriminar entre textos agresivos y no agresivos.

[12] https://amiibereval2018.wordpress.com/

2.4.3. HatEval

SemEval es un taller internacional de análisis semántico computacional que se realiza cada año. Su objetivo es explorar la naturaleza del significado; aunque el resultado es fácil de intuir para el ser humano, lo complicado es transferir la información a la computadora. El nombre de SemEval viene del SensEval que su objetivo era desambiguar el sentido de la palabra en el contexto para diferentes idiomas. Continuando con este taller, surgió SemEval en 2007, abarcando un grupo más amplio de tareas e idiomas.

Se incorporaron nuevas áreas de estudio como el análisis de sentimientos, motivando a un grupo de investigadores para obtener mecanismos con los que caracterizar textos según declaraciones subjetivas. El taller ha continuado evolucionando y ampliando con nuevas tareas para tratar los diferentes problemas del análisis semántico.

El taller ha ido evolucionando incorporando tratando las nuevas problemáticas como la que se estudia en este trabajo. Se han incorporado dos tareas que tratan sobre la detección de mensajes abusivos dentro de la edición SemEval 2019. Las dos tareas son HatEval, enfocado en la detección de *hate speech* y OffensEval que su objetivo es detectar contenido ofensivo en general; no especializado en discurso de odio.

Los datos han sido extraídos utilizando diferentes estrategias. En lo que respecta al marco temporal, la mayoría han sido extraídos desde julio de 2018 a septiembre de 2018, excepto los mensajes destinados a mujeres. La mayor parte de los mensajes de entrenamiento para las mujeres han derivado de otras colecciones previas realizadas para la identificación de misoginia [10]. Para recolectar estos tweets se han empleado diferentes técnicas:

- Monitorizar víctimas potenciales de perfiles *haters*.
- Descargar el historial de los *haters* identificados.

- Filtrar contenido de Twitter con palabras claves, hashtags...

Las palabras claves que más aparecen en las colecciones de tweets[13] son *migrant, refugee, #buildthatwall, bitch, hoe, women* para inglés e *inmigra-, "arabe", "sudaca", "puta", "callate", "perra"* para español. El *dataset* realizado por *HatEval-2019* está compuesto por 19600 *tweets* (13000 *tweets* para inglés y 6600 *tweets* para español) distribuidos en los dos objetivos de la tarea: 9091 sobre inmigrantes y 10509 sobre mujeres. A continuación, en la *Tabla 2.1* se muestra los porcentajes de distribución de cada categoría para la colección en español.

	Entrenamiento		**Evaluación**	
Etiqueta	**Inmigrante**	**Mujer**	**Inmigrante**	**Mujer**
Odio	41.93	41.38	40.50	42.00
No odio	58.07	58.62	59.50	58.00
Individual	13.72	87.58	32.10	94.94
Grupal	86.28	12.42	67.90	5.06
Agresivo	68.58	87.58	50.31	92.56
No agresivo	31.42	12.42	46.69	7.44

Tabla 2.1. Distribución de porcentajes para la colección de español

El *dataset* ha sido etiquetado mediante la plataforma *Figure Eight*[14] que es una plataforma de crowdsourcing. El esquema de anotación utilizado es es una fusión de las dos técnicas utilizadas en el desarrollo de un corpus de detección de odio [11] y misoginia [12], incluyendo las siguientes categorías:

- ***Hate speech* (HS),** valor binario que indica si existe discurso de odio contra uno de los objetivos (mujeres o inmigrantes): 1 si ocurre, 0 si no.

[13] https://github.com/msang/hateval/blob/master/keyword_set.md
[14] http://www.figure-eight.com/

- **Grupo objetivo (TR),** si ocurre *hate speech,* valor binario que indica si el mensaje de odio va destinado a una persona individual (1) o grupo genérico(0).
- **Agresividad (AG),** si ocurre *hate speech*, valor binario que indica si el tweet es agresivo (1) o no(0).

Para el etiquetado de estas clases se le ha dado una guía en español e inglés que incluye la definición de *hate speech* tomada para los dos grupos mujer e inmigrantes. El sistema de etiquetado ha reportado la confianza que le da a cada clase (*hate speech,* grupo objetivo y agresividad). En inglés para los campos HS, TR y AG sus valores de confianza son 0.83, 0.70 y 0.73 respectivamente. Para español, los valores de confianza para HS, TR y AG son 0.89, 0.47 y 0.47.

Como se puede observar, la confianza para el etiquetado en la categoría grupo objetivo y agresividad en español es bastante bajo, ya que no llega ni a 0.50. Este trabajo se ha centrado en el idioma español, por lo tanto, debido a esa confianza tan baja, solo se va a tener en cuenta la categoría *hate speech* del etiquetado proporcionado por la organización de la tarea.

HatEval (*Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter*, según su nombre en inglés) consiste en detectar contenido de odio en textos publicados en Internet, concretamente en Twitter sobre dos grupos perseguidos como son las mujeres y los inmigrantes. Es una tarea multilingüe realizada en inglés y español. La propia tarea proporciona a los participantes los *datasets* de entrenamiento y test.

La tarea HatEval se divide en dos tareas principales. La tarea A se centra en la detección de *hate speech*. La tarea B se centra en la investigación de aspectos que puedan caracterizar el tipo de odio. Para ello, en esta tarea se identifica si el odio se dirige hacia un individuo o hacia un grupo de personas; y a continuación, identificar si el contenido es ofensivo.

Se puede dividir la tarea HatEval en tres clasificaciones binarias diferentes. La primera clasificación corresponde a la detección de discurso de odio perteneciente a la primera tarea. La segunda tarea se divide en dos clasificaciones: la primera, encargada de detectar el objetivo del odio determinando si el odio pertenece a un individuo o un grupo de personas; la segunda, estudia si el texto además de discurso de odio es contenido agresivo. HatEval-2019 se puede dividir en tres subtareas de clasificación que son las siguientes:

- Tarea A: discurso de odio - no discurso de odio
- Tarea B1: individuo - grupo
- Tarea B2: agresivo - no agresivo.

A continuación, en la *sección 3.1.1.* se va a analizar cómo se han extraído los datos en este taller y las propiedades del dataset. En la *sección 3.1.2.* se van a mostrar algunos participantes con los resultados obtenidos en el taller y las técnicas utilizadas.

2.4.3.1. Sistemas participantes y resultados

HatEval ha sido uno de las tareas más populares de SemEval 2019 con un total de 108 envíos para la tarea A y 70 para la tarea B. Hubo entregas de 74 equipos diferentes, de los que 22 participaron en ambos lenguajes.

Además de los enfoques tradicionales de Machine Learning, se ha observado que más de la mitad de los participantes han investigado sobre el Deep Learning. Se ha demostrado que la mayoría de sistemas adoptados conocidos han dado buenos resultados para el tratamiento de texto desde las redes neuronales hasta modelos de lenguaje propuestos como los de Sabour et Al [13] y Cer et Al [14]. Por ello, muchos recursos externos pre entrenados como los Word Embeddings.

Muchos de los sistemas enviados en esta tarea han adoptado técnicas de preprocesamiento tradicional como son la tokenización, transformar todos los

caracteres a minúscula (conocido como *lowercase*), eliminar palabras vacías (*stopwords*), URL y signos de puntuación. Muchos de los participantes han utilizado un preprocesamiento específico para tweets como son la segmentación de *hashtags*, convertir la jerga específica y la traducción de los *emojis* a palabras.

Para la subtarea A en español, se enviaron 39 sistemas. Los equipos con mejores resultados fueron *Atalaya* y *MineriaUNAM* obteniendo un F1-score de 0.73, ambos apoyados en el *Support Vector Machines.* El equipo *Atalaya* ha estudiado sofisticados sistemas, obteniendo el mejor resultado con un SVM basado en características como bolsa de palabras, bolsa de caracteres y *tweets embeddings*, calculadas a partir de fastText basado en vectores de palabras orientado al sentimiento. El sistema propuesto por equipo de *MineriaUNAM* utilizando también un SVM de núcleo lineal, buscando la mejor configuración para combinar características de patrones lingüísticos, un lexicón de palabras agresivas y diferentes tipos de n-gramas (caracteres, palabras, POS tags, palabras agresivas y símbolos de puntuación). El equipo *MITRE* ha logrado un F1-score de 0.729, presentando un nuevo método para adaptar modelos BERT pre entrenados a datos de Twitter utilizando un corpus recolectado de esta red social en el mismo periodo que extrajo el corpus HatEval.

La siguiente tabla (*Tabla 2.2*) muestra las estadísticas básicas de los participantes de la subtarea A y subtarea B expresadas en términos de F1-score.

	Subtarea A		**Subtarea B**	
	Español	**Inglés**	**Español**	**Inglés**
Min.	0.4930	0.3500	0.4280	0.1590
Q1	0.6665	0.4050	0.5820	0.2790
Media	0.6821	0.4484	0.6013	0.3223
Mediana	0.7010	0.4500	0.6160	0.3120
StdDev	0.0521	0.0569	0.0662	0.0890

Q3	0.7165	0.4880	0.6365	0.3570
Máx.	0.7300	0.6510	0.7050	0.5700

Tabla 2.2. Estadísticas básicas de los sistemas

3. DISCURSO DE ODIO EN LAS REDES SOCIALES

En este capítulo se estudia el análisis del discurso de odio en las principales redes sociales más utilizadas en la actualidad.

Todos los días, la cantidad de usuarios activos en redes sociales es cada vez mayor, debido a que la Web 2.0 ha promovido que los usuarios no son simples receptores de información y pasan también a ser productores. Esto significa que cada vez son más las personas que están utilizando estos sitios para mantenerse en contacto con amigos, familiares y compañeros de trabajo, en lugar de usar los medios tradicionales como el correo electrónico o las llamadas telefónicas.

Según un estudio realizado en enero de 2020, la cantidad de personas que utilizan internet ha ascendido a 4540 millones de usuarios como se observa en la *Ilustración 3.1.*. Esto significa un incremento del 7% respecto al año anterior, con 298 millones de usuarios nuevos. Este aumento de usuarios en internet ha derivado también en un incremento de usuarios en redes sociales. En enero de 2020, el número de usuarios en redes sociales es de 3800 millones. Esto se traduce en un aumento del 9% respecto al 2019, con un aumento de 321 millones de usuarios.

Por otro lado, el aumento de usuarios en las redes sociales ha producido la nueva existencia de otras. Este mundo es muy cambiante, algunas redes sociales han tenido un incremento exponencial mientras que otras han llegado a su desaparición. En la *Ilustración 3.2.* se puede observar como Facebook lidera el ranking con 2449 millones de usuarios.

Los datos que se han analizado han conllevado a investigadores de distintas áreas, como el PLN, a analizar y realizar estudios en la Web 2.0 y concretamente en las redes sociales.

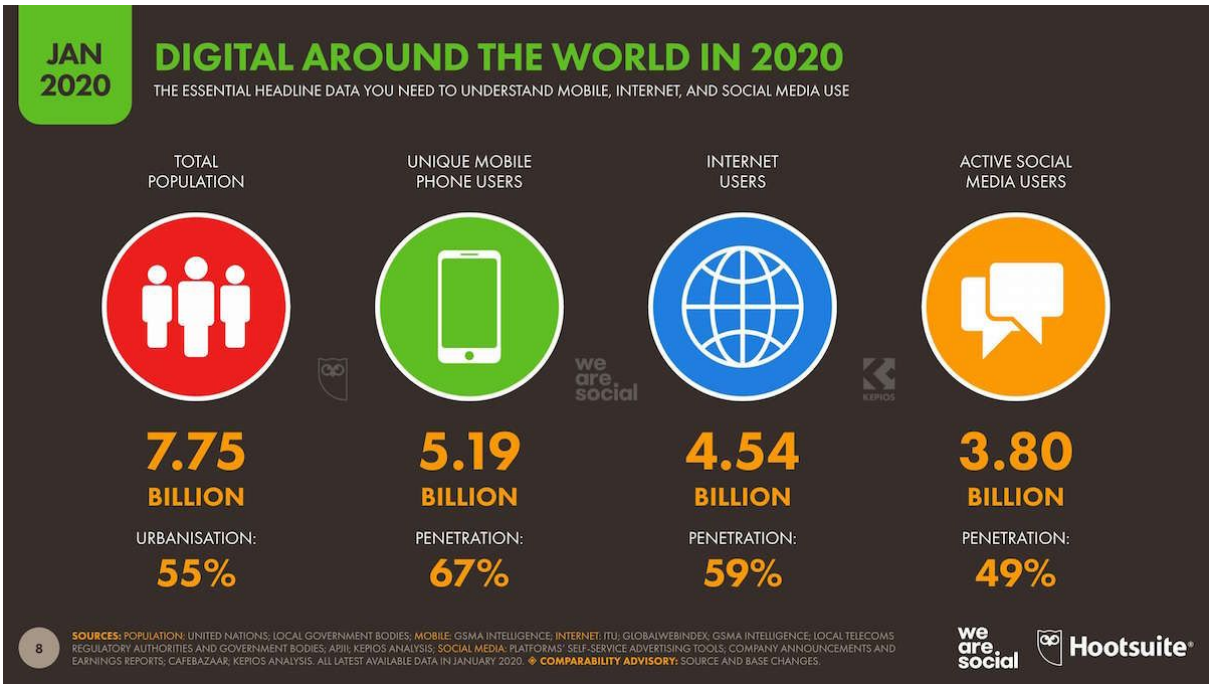

Ilustración 3.1. Uso de internet en 2020

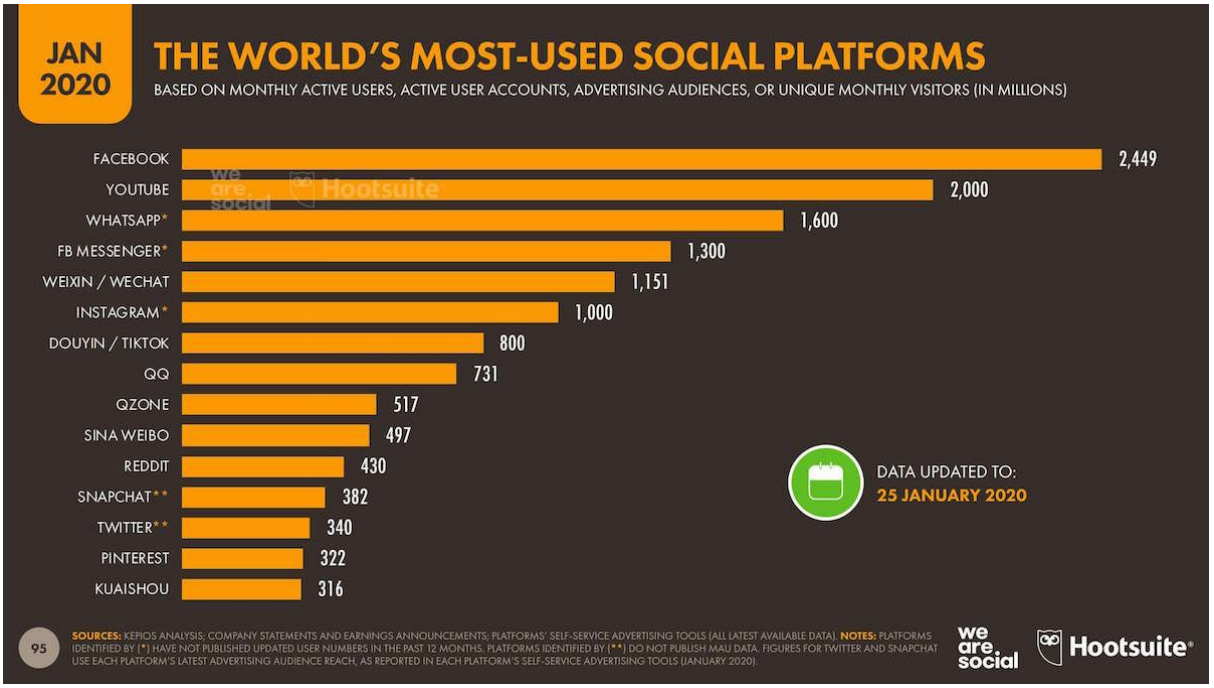

Ilustración 3.2. Ranking de redes sociales en 2020

3.1. Discurso de odio en España

En España, existe una preocupación creciente por el aumento del discurso de odio en internet y por las conductas conducidas por prejuicios y estereotipos que pueden lastrar la convivencia social. Es un punto candente, ya que trata sobre el límite de los derechos de libertad de expresión.

Esta preocupación es reciente, los delitos de odio no empiezan a tener relevancia social conocido como este concepto hasta 2009 y se empieza a trabajar las primeras estrategias sobre el discurso de odio a partir de 2015. El Ministerio del Interior introduce un nuevo apartado dedicado a los discursos de odio, aportando algunas cifras clave (Ministerio del Interior, 2017):

- Los responsables de los delitos de odio registrados, son los hombres quienes acaparan casi la totalidad de las detenciones e investigados (78.9%). El rango de edad de los responsables es:
    - De 0 a 17 años: 14.9%.
    - De 18 a 25 años: 24.7%.
    - De 26 a 40 años: 28.8%.
- Sobre los discursos de odio, en 2017 se registró un tercio más de casos que en 2015, con una mayor incidencia en los ámbitos de ideología, racismo, xenofobia y orientación sexual.
- Los principales medios de difusión de los discursos de odio son difundidos a través de Internet (36.5%) y las redes sociales (17.9%).

En 2017, el número de hechos conocidos de discurso de odio en Internet se situaba en 1.419; este número crece un 12,6% hasta llegar a 1.598 en 2018 y en 2019 se alcanzan los 1.706 casos, con un incremento del 6,8%.

3.2. Discurso de odio en Twitter

Twitter es una red social fundada en 2006. Fue concebida inicialmente como una plataforma basada en SMS. Desde entonces, Twitter se ha convertido en una de las mayores plataformas sociales y tiene más de 330 millones de usuarios activos mensualmente.

En los últimos meses, Amnistía Internacional ha llevado a cabo una investigación cualitativa y cuantitativa sobre las experiencias de las mujeres en las plataformas de los medios sociales y el impacto de la violencia y el abuso dirigidos a las mujeres en Twitter, con especial atención en el Reino Unido y Estados Unidos. Twitter ha ido evolucionando sus reglas de conducta de odio con el objetivo de proporcionar a todos los usuarios la capacidad de crear y compartir ideas e información, y expresar sus opiniones y creencia, sin ningún tipo de obstáculos siempre que no fomente la violencia contra otras personas ni atacarlas o amenazarlas por motivo de su raza, origen étnico, nacionalidad, orientación sexual, género, religión, edad, discapacidad o enfermedad grave.

Estudios e iniciativas anteriores han abordado la monitorización del discurso del odio online en diferentes países, sin embargo, no se aprecia una propuesta articulada de cómo abordar los mensajes de odio online. Desde las instituciones públicas y empresas privadas (principalmente redes sociales) se está empezando a realizar un esfuerzo por comprender este fenómeno y combatirlo con un enfoque común.

4. DESARROLLO DEL PROYECTO

En este capítulo se explica el desarrollo software del proyecto. En primer lugar, se encuentra la explicación del problema a resolver junto a cómo resolverlo. Se indica la metodología a seguir, las historias de usuario, la propuesta de solución y las herramientas utilizadas para la implementación del sistema.

4.1. Descripción del problema

Teniendo en cuenta el propósito y los objetivos del proyecto, ya presentados y tras mostrar el estado del arte de los análisis de detección de discurso de odio, se procede a explicar con detalle la descripción del problema.

Con la aparición de la web 2.0, ha crecido exponencialmente la cantidad de información que se produce en las redes sociales. Entre la información que ha aumentado, también ha incrementado el discurso de odio en las redes sociales. Estas compañías están aumentando sus recursos para detectar estos mensajes para eliminarlos y sancionar a estos usuarios. El discurso de odio se emplea concretamente en los grupos sociales más vulnerables como las mujeres y los inmigrantes.

Por este motivo, surge la necesidad de realizar un sistema que sea capaz de detectar el discurso de odio en español.

4.2. Objetivos del sistema

El objetivo principal de este sistema es analizar y detectar si un contenido contiene discurso de odio, además de dar al usuario una herramienta que facilite el uso del sistema.

Este objetivo se puede descomponer en:

- Objetivo 1: Sistema de detección discurso del odio. Elaborar un sistema que sea capaz de detectar si existe discurso de odio en un texto.

- Objetivo 2: Búsqueda de la respuesta. A partir de una consulta introducida por un usuario devolver una respuesta.
- Objetivo 3: Presentación de la respuesta. Mostrar la respuesta de la forma más clara y concisa al usuario.

4.3. Metodología del desarrollo del software

La metodología del desarrollo del software es una disciplina o área de la Informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en los sistemas de información.

4.3.1. Comparativa metodología ágil y tradicional

La metodología utilizada para este proyecto es la metodología ágil. A continuación, se exponen las diferencias entre una metodología y otra.

La metodología tradicional busca “imponer” disciplina al proceso de desarrollo del software para hacerlo predecible y eficiente. Estas metodologías tienen un enfoque predictivo, donde se sigue un proceso secuencial que no tiene marcha atrás. La estimación y planificación se realiza una sola vez en el inicio del proyecto. Por ello, la estimación tiene una gran importancia en el proyecto ya que de ella dependen los recursos que se utilizarán en el proyecto.

Se puede observar que la metodología tradicional se puede aplicar cuando se tiene mucha experiencia en un determinado tipo de producto para poder estimarlo. También puede funcionar para proyectos con los requisitos muy definidos en un entorno estable.

Ilustración 4.1. Fases de la metodología tradicional

La metodología ágil se caracteriza por ser adaptativa y flexible. Esto significa que recoge con normalidad los cambios debido a que son eventos esperados y se acogen con normalidad. Las metodologías ágiles más conocidas son Scrum, Kanban y Lean. Para que esta metodología funcione correctamente debe ser continúa la comunicación con el cliente, por si hay necesidad de cambio de requisitos o funcionalidades en cualquier momento. El cliente puede ser un miembro más del equipo. El grupo de trabajo debe ser un número reducido de trabajadores que estén motivados, la cohesión del grupo tiene una gran importancia.

Los proyectos ágiles, se fragmentan en pequeños proyectos de un período corto de tiempo denominados *sprints*.

Una de las máximas de las metodologías ágiles es la de anteponer el producto sobre las tareas. El objetivo es tener un producto funcional lo antes posible. En las primeras instancias del proyecto ya existe un producto que ofrecer, que va mejorando y ampliándose en cada *sprint*.

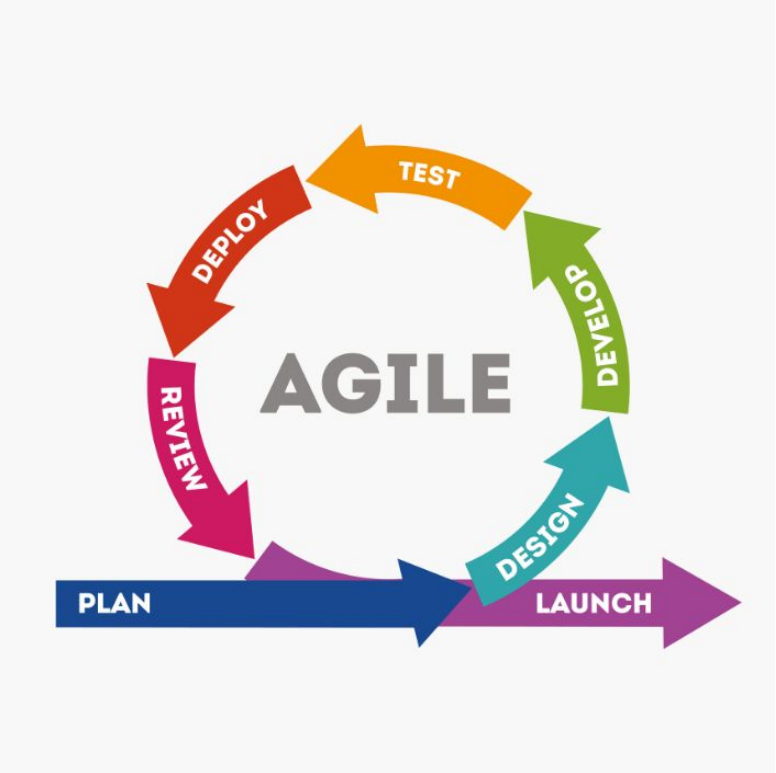

Ilustración 4.2. Fases de la metodología ágil

Una vez comparadas las dos metodologías, se ha elegido una metodología ágil por varias razones:

- Se va a desarrollar el proyecto por iteraciones.
- Al ser un proyecto de investigación, es complicado tener los requisitos bien definidos durante el proyecto.
- Se irá teniendo contacto periódicamente con los tutores (simulando al cliente).
- Pueden surgir cambios durante el proyecto.

4.3.2. Metodología ágil

Las metodologías ágiles que más se utilizan en la actualidad son las siguientes:

- SCRUM
- Extreme Programming

- Agile Inception
- Kanban

4.3.2.1. SCRUM

Se basa en una estructura de desarrollo incremental. Cualquier ciclo de desarrollo del producto se fragmenta en “pequeños proyectos” divididos en distintas etapas: análisis, desarrollo y *testing.* En la etapa de desarrollo se encuentra lo que se conoce como interacciones de entrega o *sprint*, que son las entregas parciales del producto.

Esta metodología permite abordar proyectos complejos con una gran flexibilidad y rapidez. La estrategia irá orientada a gestionar y normalizar los errores que se puedan producir en desarrollos demasiado largos, a través de frecuentes reuniones con el cliente para seguir los objetivos establecidos.

Estas reuniones son el pilar fundamental en el que se basa la metodología SCRUM. Podemos diferenciar las reuniones en reuniones de planificación, reunión diaria también conocida como *daily*, reunión de revisión y reunión de retrospectiva. La reunión de retrospectiva es una de las más importantes, ya que se realiza después de terminar un *sprint* para reflexionar y proponer nuevas mejoras en el proyecto.

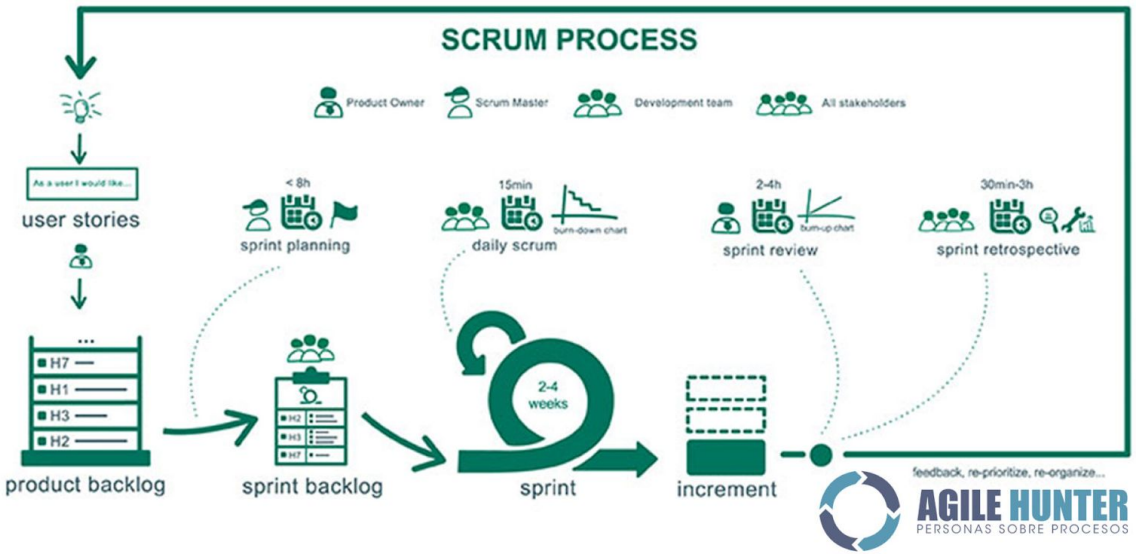

Ilustración 4.3. Fases de la metodología SCRUM

4.3.2.2. Extreme Programming

El objetivo principal de *Extreme Programming (XP)* es ayudar la relación entre clientes y empleados. Esto sería muy útil para *startups* o empresas en proceso de consolidación. La clave de *Extreme Programming* es potenciar las relaciones personales, potenciando el trabajo en equipo fomentando la comunicación. Sus principales fases son:

- Planificación del proyecto con el cliente.
- Diseño del proyecto.
- Codificación. Los programadores trabajan en parejas para obtener resultados más eficientes y de calidad.
- Pruebas.

4.3.2.3. Agile Inception

Agile Inception está orientada a la definición de los objetivos generales de las empresas, clarificando cuestiones como el tipo objetivo, propuestas de valor añadido o las formas de venta. Su entorno gira a través del método *elevator pitch*,

que consiste en pequeñas reuniones entre los socios y equipo de trabajo con intervenciones con límite de tiempo.

4.3.2.4. Kanban

Kanban ha sido la metodología elegida para el desarrollo de software de este proyecto. La metodología Kanban consiste en la elaboración de un cuadro para ordenar las tareas. Este cuadro de tareas tiene tres columnas: pendientes, en proceso y terminadas. Todos los miembros del equipo deben de tener acceso a este cuadro, para que no se dupliquen las tareas y puedan ver el estado del proyecto ayudando a la mejora de productividad y eficiencia del trabajo. Las ventajas que proporciona esta metodología son:

- Planificación de tareas.
- Mejora en el rendimiento de trabajo de equipo.
- Métricas visuales.
- Plazos de entrega continuos.

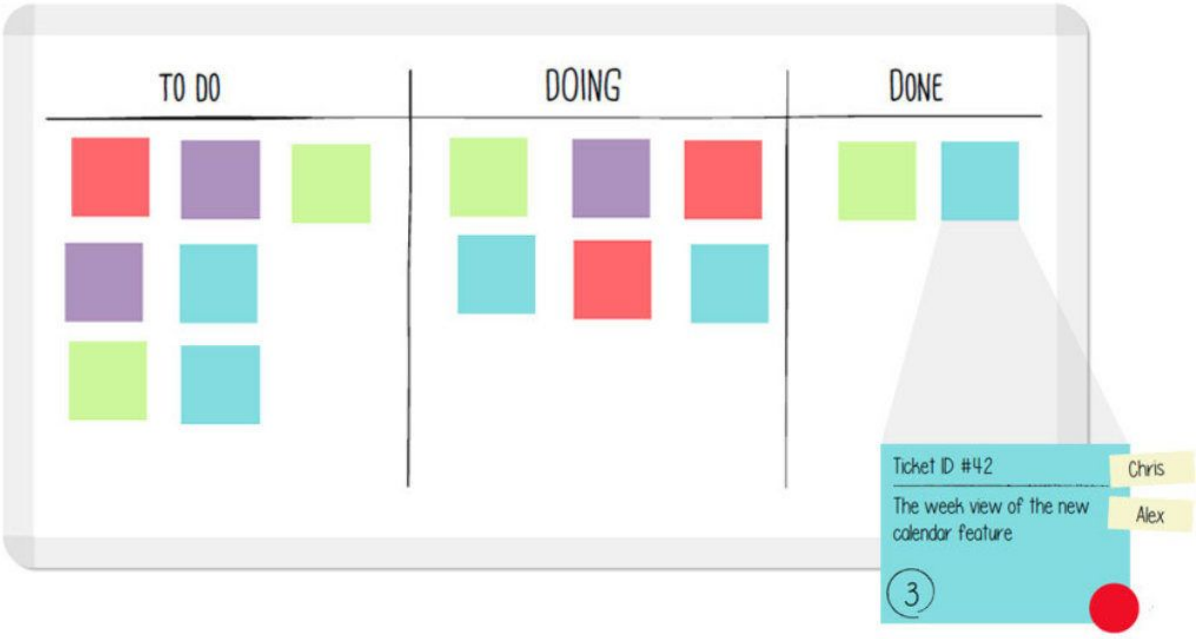

Ilustración 4.4. Fases de la metodología Kanban

4.4. Historias de usuario

Las historias de usuario son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir los requisitos sobre nuestro software. Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea esa nueva capacidad, generalmente es el cliente o usuario del sistema.

Una buena historia de usuario debe de seguir el modelo INVEST: independiente, negociable, estimable, pequeña (small) y *testeable*. A continuación, se describen las características:

- **Independiente:** una historia debe ser lo más autónoma posible. Una dependencia con otra historia hará más difícil la planificación, priorización y estimación.
- **Negociable:** la tarjeta debe ser una una descripción corta sin detalles. Los detalles se acordarán en la reunión con el cliente.
- **Valiosa:** cada historia tiene que tener valor para el cliente.
- **Estimable:** una tarjeta tiene que poder ser estimable, esto permitirá que se puedan priorizar y planificar. El problema en la estimación puede venir por falta de conocimiento en el dominio o que la tarea sea demasiado general.
- **Pequeña:** una historia debe ser concreta y no tener una duración muy extensa. Al ser una tarea más simple habrá menos desviación.
- **Testeable:** la tarea debe de poder probarse. Si no se puede probar, no se podrá confirmar que la tarea está bien finalizada.

Una historia de usuario se representa:

1. **ID:** identificador de la tarea.
2. **Título:** asunto de la historia de usuario. Debe de ser descriptivo.
3. **Descripción:** se emplea la estructura “Como… Quiero… Para …”.

4. **Estimación:** valoración del coste de la historia de usuario. La unidad de tiempo utilizada en Kanban son semanas.
5. **Prioridad:** preferencia de la historia de usuario respecto al resto de historias de usuario del proyecto. La prioridad se valora en: baja, media, alta, crítica.
6. **Dependencias:** historias de usuario previas necesarias para realizar la historia de usuario actual.
7. **Criterios de aceptación:** pruebas para comprobar si la historia de usuario se ha finalizado con éxito.

La plantilla que se va a utilizar para representar las historias de usuario se muestra en la *Tabla 4.1.*:

ID	**Título**	
Descripción		
Estimación		Prioridad
Dependencias:		

Tabla 4.1. Plantilla historia de usuario

Las historias de usuario del presente proyecto son las siguientes:

1	**Sistema preprocesamiento del texto**	
Como usuario quiero tratar el texto previo al detector de odio para que el contenido siempre tenga la misma presentación		
1 semana		Crítica
Dependencias: Ninguna		

Tabla 4.2. Historia de usuario "Sistema preprocesamiento texto"

2	**Reconocimiento de odio en tweets**	
Como usuario quiero reconocer si existe discurso de odio en los tweets descargados.		
7 semanas		Crítica
Dependencias: 1		

Tabla 4.3. Historia de usuario "Reconocimiento de odio en tweets"

3 Validación sobre el reconocedor de discurso de odio	
Como usuario quiero que el sistema supere unas métricas estadísticas para validar la calidad del sistema	
1 semana	Alta
Dependencias: 1,2	

Tabla 4.4. Historia de usuario "Validación sobre el reconocedor de discurso de odio"

4 Monitor discurso de odio	
Como usuario quiero una interfaz para poder interactuar con el sistema.	
4 semanas	Alta
Dependencias: 2	

Tabla 4.5. Historia de usuario "Monitor de discurso de odio"

5 Tipos interacción con monitor	
Como usuario quiero poder interactuar de diferentes formas con el monitor de discurso de odio	
2 semanas	Media
Dependencias: 4	

Tabla 4.6. Historia de usuario "Tipos interacción con monitor"

Además de las historias de usuario descritas anteriormente, es fundamental incluir las historias de usuario no funcionales. Son aquellas relacionadas con el rendimiento, tiempo de carga, disponibilidad, étc. Pese a que no son funcionalidades del sistema es importante cumplirlas para que la experiencia de usuario sea exitosa. Dado que estas historias de usuario se van cumpliendo a medida que se desarrolla el proyecto no se convierte en incremento del tiempo de desarrollo del proyecto.

6	**Tiempo de respuesta**
Como usuario quiero que la interfaz responda a cualquier petición en el tiempo estimado	
0 semanas	Baja
Dependencias: 4,5	

Tabla 4.7. Historia de usuario "Tiempo de respuesta"

7	**Disponibilidad**
Como usuario quiero que la aplicación esté disponible durante el número de horas estimadas durante el año	
0 semanas	Baja
Dependencias: 4	

Tabla 4.8. Historia de usuario "Disponibilidad"

8	**Accesibilidad multiplataforma**
Como usuario quiero el monitor de discurso de odio pueda ser utilizado desde los diferentes navegadores	

0 semanas	Media
Dependencias: 4	

Tabla 4.9. Historia de usuario "Accesibilidad multiplataforma"

4.5. Planificación software

Para la planificación del proyecto se ha utilizado la herramienta Trello[15] para seguir la metodología Kanban. Trello sirve para gestionar tareas permitiendo organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuesto por listas de tareas para simplificar cada tarjeta. Para mejorar la rutina de trabajo, la plataforma permite crear prioridades, tiempos y avisos.

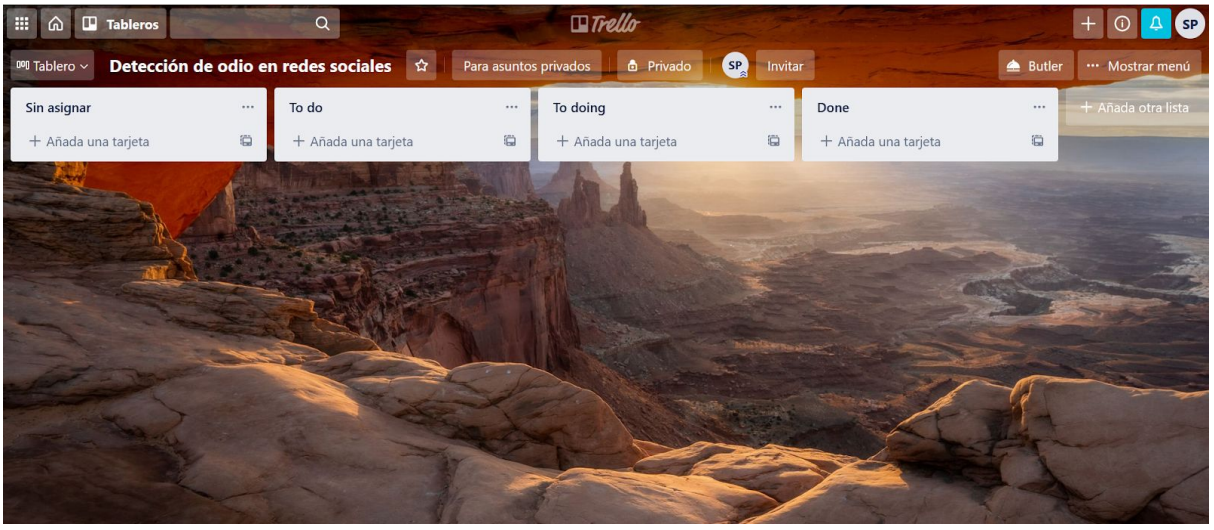

Ilustración 4.5. Tablero Kanban

Como se puede observar en la *Ilustración 4.5.* El tablero de este proyecto está formado por cuatro columnas: “Not assigned”, “To do”, “To doing”, “Done”. Cada tarjeta representará una historia de usuario. Cuando se crea una tarea se sitúa en la primera columna “Not assigned” hasta que se le asigne un desarrollador. Al ser un Trabajo Fin de Máster sólo hay un desarrollador, pero las tareas se asignarán a los diferentes roles de la *Sección 1.6. Estimación de costes.* Una vez la tarea pertenece a un desarrollador pasará a la segunda columna “To do”, se encuentran las tareas que tienen un responsable, pero no se han comenzado. Al empezar una tarea, la tarjeta se mueve a la tercera columna “To doing” dónde se

[15] https://trello.com/

encuentran las tareas en progreso. Una vez se termina una tarea, se mueve a la última columna “Done”.

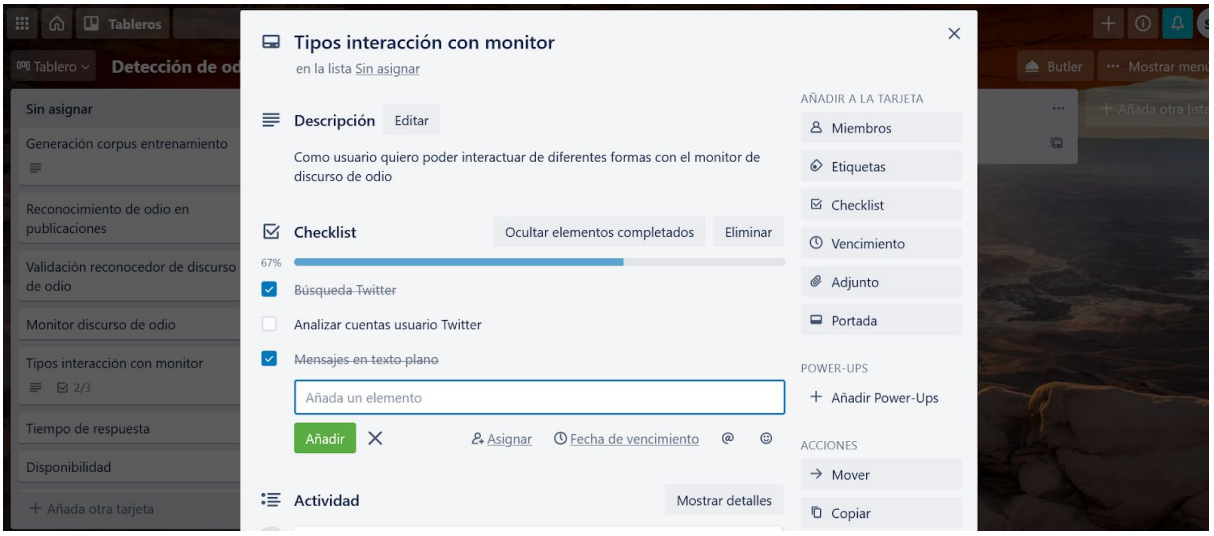

Ilustración 4.6. Parámetros historia de usuario en Kanban

A cada tarjeta se le puede añadir una serie de parámetros que se observan en la *Ilustración 4.6.* La fecha de vencimiento representa la fecha estimada de finalización de la tarea para poder estimar si el proyecto va con retraso. En el apartado *Checklist* se añaden las tareas en la que se simplifica la tarjeta.

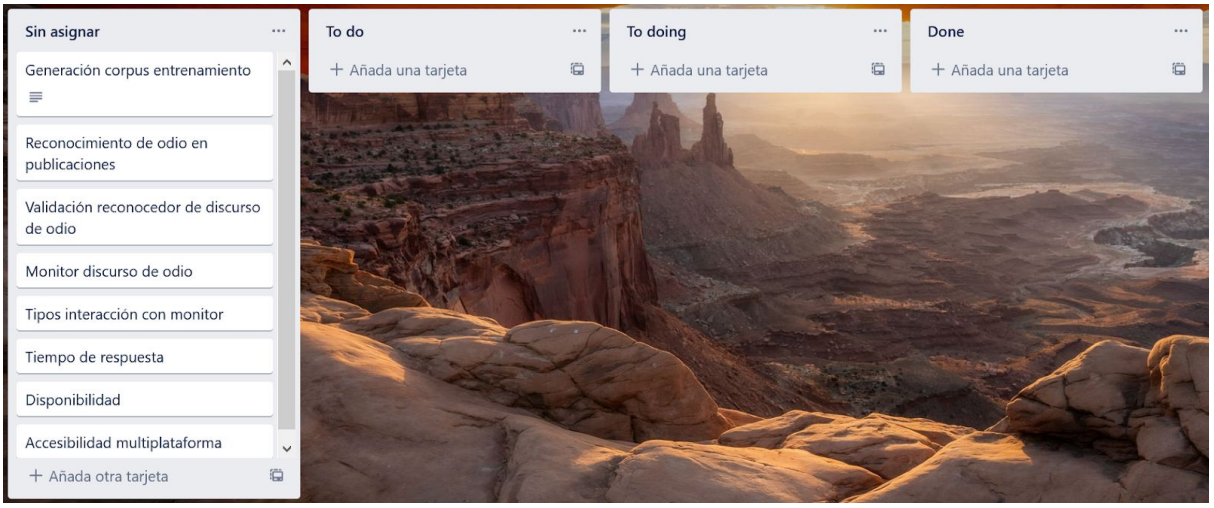

Ilustración 4.7. Estado inicial tablero Kanban

4.6. Propuesta de solución

La solución propuesta en este Trabajo Fin de Máster consiste en el diseño y desarrollo de un sistema detector de contenido con discurso de odio. Los módulos elementales son:

- Sistema de recopilación de mensajes desde la red social *Twitter*.
- Sistema de identificación de discurso de odio.
- Interfaz web donde el usuario pueda interactuar para hacer diferentes consultas.

4.7. Descripción de la solución

En este apartado se explica el desarrollo de la solución del proyecto. Cada iteración representa una historia de usuario citadas en la *Sección 4.4.* A continuación, se describe el desarrollo de cada una de las iteraciones.

4.7.1. Iteración 1

Iteración 1: "Como usuario quiero tratar el texto previo al detector de odio para que el contenido siempre tenga la misma presentación".

La iteración 1 se corresponde a la historia de usuario "*Sistema preprocesamiento del texto*" mostrada en la *Tabla 4.2.*

En el ámbito de la detección de discurso de odio tiene una gran importancia el tratamiento de texto previo al sistema clasificador. Para realizar este preprocesamiento se ha utilizado principalmente la herramienta Ekphrasis[16]. Ekphrasis permite las siguientes funcionalidades:

- Tokenizador orientado a las redes sociales (Facebook, Twitter…). Tiene un tratamiento especial para emoticonos, *emojis* y otras expresiones no estructuradas como fechas, horas…

[16] https://github.com/cbaziotis/ekphrasis

- Segmentado de palabras que divide una larga cadena en sus palabras constituyentes. Tiene una gran importancia para simplificar los *hashtags*.
- Corrector ortográfico que reemplaza las palabras mal escritas por la candidata con más probabilidad.

Las técnicas utilizadas finalmente en el preprocesamiento de texto son las siguiente:

- Eliminación de *stopwords*.
- Convertidos todos los caracteres a minúscula.
- Sustitución de menciones de usuario por *"user"*. Se ha decidido que para el procesamiento no es importante el nombre del usuario, sino el hecho de haber mencionado a un usuario.
- Sustitución de urls por *"url"*. Se realiza un procesamiento similar a las menciones de usuario.
- Eliminación de signos de puntuación y acentos.

A continuación, se muestran diferentes ejemplos, del texto antes de procesar y como queda al pasar por el sistema desarrollado.

[
{
"text": "Easyjet quiere duplicar el numero de mujeres piloto Veras tu para aparcar el avion.. http://t.co/46NuLkm09x",
"text_preprocess": "easyjet quiere duplicar numero mujeres piloto veras aparcar avion repeated url"
},
{
"text": "@MaivePerez Llora te lo.mereces por zorra",
"text_preprocess": "user llora lomereces zorra"
},
{

"text": "Todos: -#NoTodosLosHombres -PPK, presidente de lujo -Peru no va al mundial 2017: Ja, ilusos.",

"text_preprocess": "hashtag hombres hashtag ppk presidente lujo peru va mundial number ja ilusos"

},

]

Estas técnicas de preprocesamiento son las seleccionadas para el sistema de preprocesamiento. Estas técnicas elegidas, se han auto alimentado con los resultados proporcionados por el sistema de la *Iteración 2*. Finalmente, se realizan las técnicas que más benefician al reconocedor de odio de publicaciones.

4.7.2. Iteración 2

Iteración 2: "Como usuario quiero reconocer si existe discurso de odio en los tweets descargados"

La iteración 2 se corresponde a la historia de usuario "*Reconocimiento de tweets en publicaciones*" mostrada en la *Tabla 4.3.*

Una vez se ha realizado el procesamiento previo del texto, el siguiente paso es desarrollar el sistema reconocedor de discurso de odio. Para generar este sistema se crea un modelo con el *dataset* de entrenamiento. Una vez se tiene el sistema, se realiza el mismo proceso con el *dataset* de test. Se realizan diferentes ejecuciones con diferentes configuraciones buscando el mejor resultado del sistema.

En primer lugar, se extraen las diferentes características del texto para incorporarlas al clasificador. Las características *positive_words* y *negative_words* corresponden al contador de palabras positivas y negativas en el texto. Se cuenta si la palabra aparece en los lexicones utilizados para extraer palabras que indican

polaridad, para contar una palabra debe aparecer en los lexicones proporcionados por iSOL [16].

Para la extracción de emociones se utiliza una técnica similar. Las características extraídas son *alegria_words, enojo_words, miedo_words, repulsion_words, sorpresa_words* y *tristeza_words*. Estas características son extraídas del lexicón[17] SEL [17], que se trata de un diccionario compuesto por 2036 en español etiquetadas con las seis emociones de Ekman, que son las más utilizadas en el PLN.

Para el cálculo de características relacionadas con la detección de odio en el texto se ha utilizado el recurso *Hate Speech Spanish Lexicon* que consta de cuatro lexicones: palabras xenófobas, misoginia, insultos, palabras contra inmigrantes y palabras tóxicas. Las características extraídas son *xenophobia_words, misogyny_words, insults_words, immigrant_words* y *hate_words*. Se cuentan las palabras del texto que aparezcan en los diferentes diccionarios.

En el mundo de internet, el uso de mayúsculas está relacionado a simular que se está gritando. Esta característica es útil para la detección de odio ya que el uso de mayúsculas en discusiones y crispaciones. Por ello, se ha extraído el número de características que hay en mayúscula y la ratio de caracteres en mayúscula. Estas características son *upper_count* y *upper_rate*.

Otra de las características utilizadas son los *embeddings*. Esto consiste en sustituir cada palabra por un vector. Estos vectores almacenan la información semántica de la palabra, pudiendo relacionar esta palabra con otras con significado similar.

La última característica extraída por el sistema consiste en analizar la estructura del texto. En vez de analizar el contenido del texto, se hace un análisis

[17] http://www.cic.ipn.mx/~sidorov/

sintáctico. Esta medida conocida como TF-IDF (*Term Frequency times Inverse Document Frequency* en inglés) trabaja con gramas como medida, que indica la unidad mínima del documento. Se pueden seleccionar diferente cantidad de gramas, si se analiza la frase “el perro rojo” con un grama extraería “perro” y “rojo” por separado; si en cambio se analiza por bigrama, extraería “perro rojo”.Para la obtención de esta característica, el primer paso del sistema de detector de odio es calcular el TF-IDF para cada palabra del dataset. El TF-IDF es una métrica que indica qué importancia tiene ese grama o conjunto de gramas en la colección. En este caso el TF-IDF que mejor resultado está dando al sistema es unigramas y bigramas de palabras. También se calculará el TF-IDF pero en vez de a las palabras, se hará con gramas de caracteres. El mejor resultado en el sistema es cuando se ha trabajado con bigramas y trigramas de caracteres. También se ha extraído el TF-IDF al resultado obtenido por el POS tagger.

Una vez ya se han obtenido todas las características que necesita el sistema para detectar el odio, hay que entrenar un modelo. Para ello, se ha seleccionado el algoritmo SVM (*Support Vector Machine)* que es uno de los que mejores resultados está dando en la detección de odio*.* SVM es un conjunto de algoritmos de aprendizaje supervisado para la resolución de problemas de clasificación o regresión. Este algoritmo será capaz de aprender los patrones de los diferentes ejemplos del conjunto de entrenamiento que se le proporcione. Es un clasificador lineal que se basa en encontrar el margen máximo entre las dos clases a partir de unos vectores determinados llamados vectores de soporte. SVM va a generar un hiperplano que permita dividir el espacio de características en dos regiones completamente disjuntas.

4.7.3. Iteración 3

Iteración 3: “Como usuario quiero que el sistema supere unas métricas estadísticas para validar la calidad del sistema”

Para la evaluación de los sistemas de clasificación se suele utilizar varias métricas para evaluar el rendimiento del sistema, las métricas a utilizar son las siguientes:

- Precisión: sirve para medir la calidad del sistema. Indica qué ratio de los mensajes detectado como una clase que han detectado correctamente.
- Recall: indica el ratio de los mensajes que tendría que haber detectado cuantos ha detectado.
- F1: combina la medida de precisión y de *recall*. Hace más fácil comparar el rendimiento combinado de la precisión y exhaustividad.

Las fórmulas para calcular estas medidas de evaluación son las siguientes:

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN} \quad F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

dónde *TP* son los documentos extraídos como positivos que realmente son positivos,

FP son los documentos extraídos como positivos, pero realmente son negativos y,

FN son los documentos extraídos como negativos, pero realmente son positivos.

En primer lugar, se van a comparar los resultados para los diferentes sistemas alternando las características citadas anteriormente con las añadidas por el TF-IDF. En la *Tabla 4.10* se puede observar que las características que mejoran el sistema son “Pos Tag” y “Word Embeddings”. La característica “iSOL” mejora la precisión para los textos que no son de odio. Estos resultados son aproximados, ya que pueden variar dependiendo del resto de procesamiento.

Característica	Clase	Precisión	Recall	F1	F1 acc.

Sin característica	No odio	0.62	0.75	0.68	0.66
	Odio	0.70	0.57	0.63	
iSOL	No odio	0.63	0.75	0.68	0.66
	Odio	0.70	0.57	0.63	
SEL	No odio	0.62	0.75	0.68	0.66
	Odio	0.70	0.57	0.63	
SEL con confianza	No odio	0.62	0.75	0.68	0.65
	Odio	0.70	0.57	0.62	
Lexicón Hate	No odio	0.61	0.74	0.67	0.65
	Odio	0.70	0.56	0.62	
Ratio mayúsculas	No odio	0.63	0.75	0.69	0.66
	Odio	0.70	0.57	0.63	
Pos TAG	No odio	0.66	0.75	0.70	0.67
	Odio	0.69	0.59	0.64	
Word embeddings	No odio	0.63	0.76	0.69	0.67
	Odio	0.71	0.58	0.64	

Tabla 4.10. Tabla de resultados del sistema al añadir características

En segundo lugar, se va a mostrar los resultados de combinar TF-IDF de palabras con el TF-IDF de caracteres variando su número de n-gramas sin añadir las características anteriores. En la *Tabla 4.11* se puede observar que el mejor resultado para nuestro sistema es utilizando TF-IDF de 1 a 3 gramas de palabras unido al TF-IDF de 2 a 3 gramas de caracteres.

TF pal.	TF car.	Clase	Precisión	Recall	F1	F1 acc.
1-1	No	No odio	0.62	0.75	0.68	0.66
		Odio	0.70	0.57	0.63	
1-2	No	No odio	0.69	0.79	0.73	0.71

		Odio	0.74	0.62	0.68	
1-3	No	No odio	0.73	0.78	0.72	0.72
		Odio	0.70	0.65	0.67	
1-3	2-2	No odio	0.73	0.78	0.75	0.72
		Odio	0.70	0.65	0.67	
1-3	2-3	No odio	0.73	0.78	0.75	0.72
		Odio	0.70	0.65	0.67	
2-2	No	No odio	0.73	0.73	0.73	0.68
		Odio	0.60	0.61	0.61	
2-2	2-3	No odio	0.74	0.72	0.73	0.68
		Odio	0.60	0.61	0.61	
3-3	No	No odio	0.28	0.76	0.41	0.52
		Odio	0.87	0.46	0.60	

Tabla 4.11. Tabla de resultados con diferentes TF-IDF

Por último, se van a combinar los mejores sistemas de la *Tabla 4.10* con la *Tabla 4.12* para decidir el sistema final. Por lo tanto, se va a combinar el TF-IDF de 1 a 3 gramas de palabras con TF-IDF de 2 a 3 gramas de caracteres, con las diferentes características citadas anteriormente.

Característica	Clase	Precisión	Recall	F1	F1 acc.
iSOL	No odio	0.74	0.78	0.76	0.72
	Odio	0.70	0.65	0.67	
Pos TAG	No odio	0.73	0.78	0.75	0.72
	Odio	0.71	0.65	0.68	
iSOL + Pos TAG	No odio	0.73	0.78	0.76	0.72
	Odio	0.71	0.65	0.68	

Tabla 4.12. Tabla de resultados combinación TD-IDF y características

Se puede observar en la *Tabla 4.12* que la concatenación del TF-IDF de caracteres con el TF-IDF de palabras tiene tanto peso en el sistema, que añadirle características adicionales apenas varía el sistema. Por ello, se ha decidido utilizar en el monitor de discurso de odio el sistema sin añadirle estas características adicionales para ahorrar recursos del sistema y tiempo de ejecución.

Sistema	**F1-score**
Atalaya	0.73
MineriaUNAM	0.73
MITRE	0.729
CIC-2	0.727
GSI-UPM	0.725
Sistema actual en desarrollo	0.72

Tabla 4.13. Comparación resultados SemEval 2019

En la *Tabla 4.13* se puede observar que el sistema desarrollado ha obtenido un rendimiento similar a los mejores sistemas presentados en SemEval 2019 si analizamos la *F1-score*. Por lo que podemos concluir que se ha conseguido un sistema capaz de detectar el discurso de odio con una fiabilidad bastante alta.

4.7.4. Iteración 4

Iteración 4: “Como usuario quiero una interfaz para poder interactuar con el sistema”

La iteración 4 se corresponde a la historia de usuario “*Monitor discurso de odio*” mostrada en la *Tabla 4.5*

Con la finalidad de que el usuario interactúe con el reconocedor de discurso de odio, se ha decidido desarrollar un monitor de discurso de odio. Esta aplicación web va a permitir al usuario realizar consultas y la web presentar la información de la forma más sencilla y concisa.

La arquitectura a seguir para desarrollar el sitio web es el patrón Modelo-Vista-Controlador (MVC). El MVC separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos (*Ilustración 4.8*). Los elementos que componen esta arquitectura es la siguiente:

- Modelo: es la capa donde se trabajan los datos, por lo que tendrá mecanismos para acceder a la información y actualizar su estado.
- Vista: es la capa donde se muestra la información. La capa vista necesitará tener acceso a los datos para mostrar la información, por ello se solicitará la información de la capa modelo.
- Controlador: sirve de enlace entre la capa modelo y vista, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación.

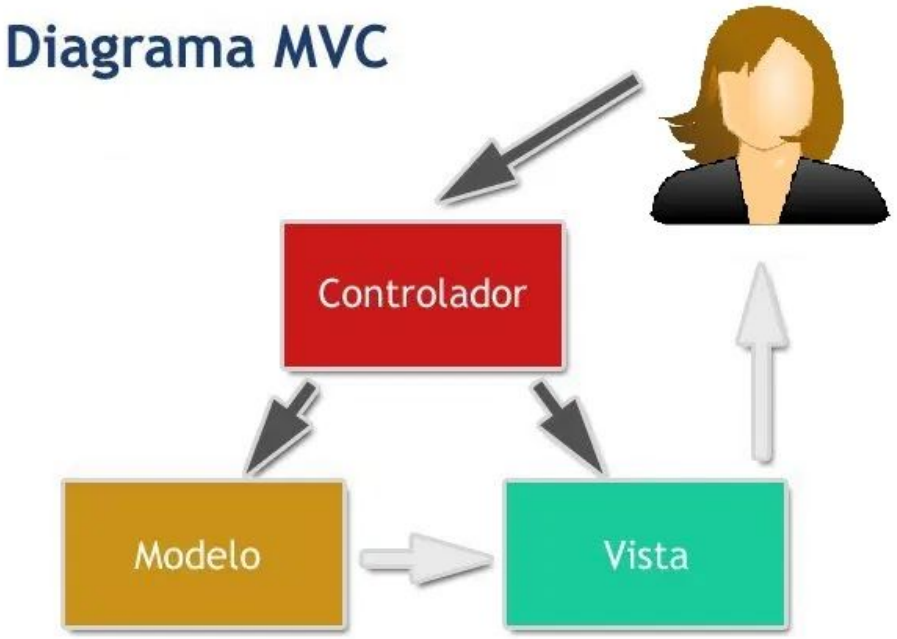

Ilustración 4.8. Arquitectura Modelo-Vista-Controlador

Para el desarrollo de la web se ha utilizado el framework Dash de Python. Dash es una biblioteca de Python basada en Flask, Plotly y RectJS. El diseño de la web se ha decantado por una web simple y minimalista, que permita al usuario centrarse en analizar la información de los usuarios. A continuación, se explica detalladamente la página web.

Para comenzar, hay que analizar los requisitos de negocio para describir la funcionalidad del sistema. En la *Ilustración 4.9* se pueden observar el diagrama de caso de uso.

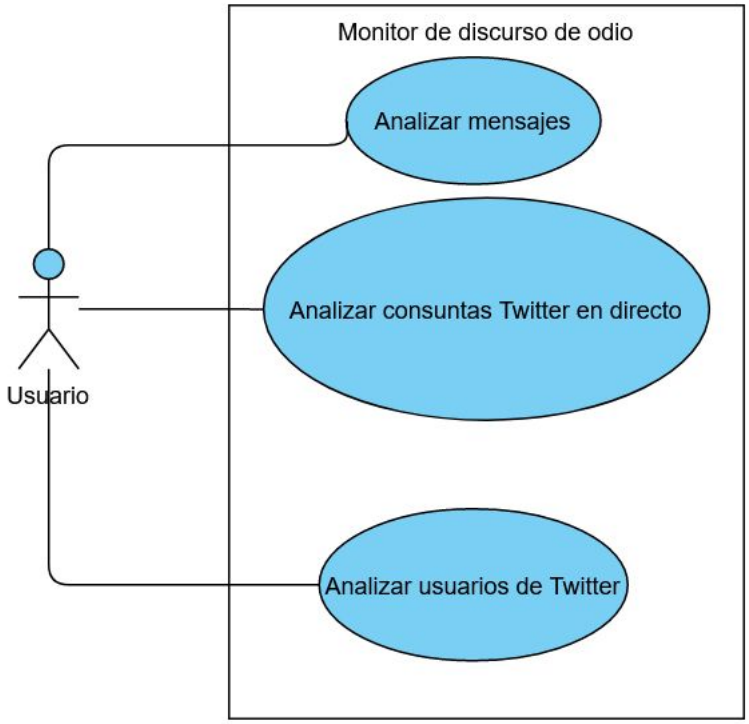

Ilustración 4.9. Diagrama de caso de uso

En segundo lugar, para implementar el patrón MVC es necesario establecer los pasos de mensaje entre las diferentes entidades. El paso de mensajes se puede observar en el diagrama de secuencia de la *Ilustración 4.10.*

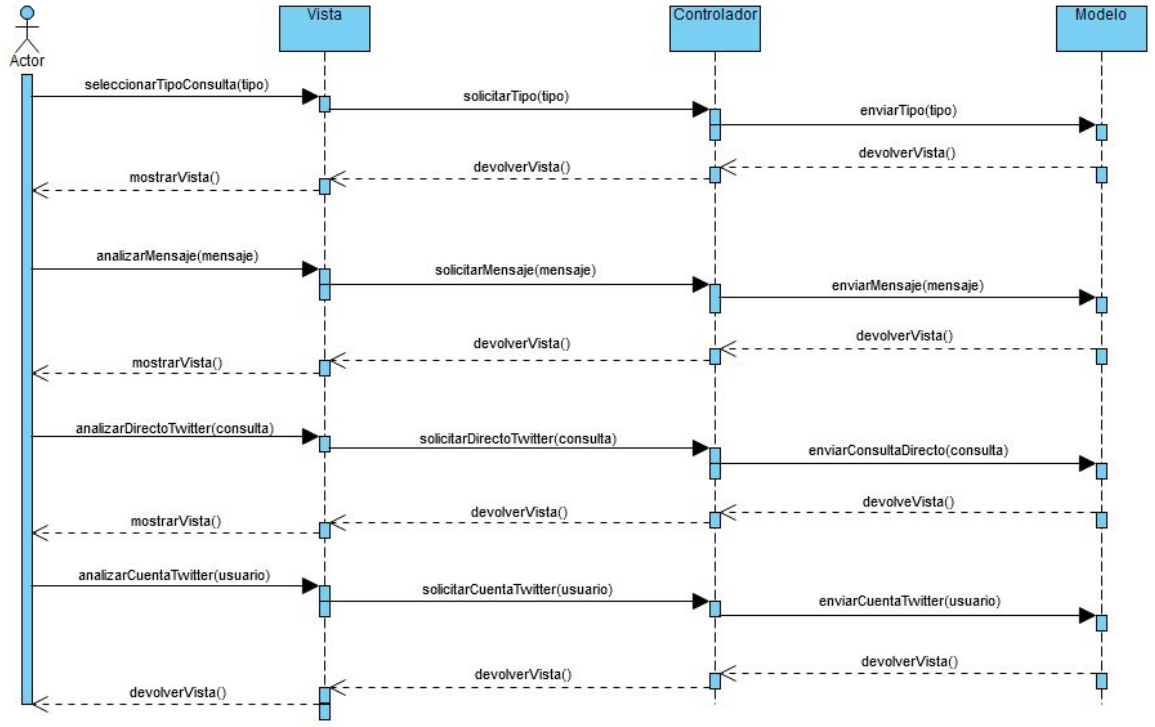

Ilustración 4.10. Diagrama de secuencia

En la *Ilustración 4.10.* se puede observar el estado inicial de la aplicación. La web está compuesta por un elemento *tab* de HTML. Las tres secciones son “Analizar mensajes”, “Directo Twitter” y “Analizar usuario Twitter”. En las tres secciones la web tendrá una estructura similar. En la parte superior, hay una entrada de texto para que el usuario realice su consulta. En la parte inferior a la entrada de texto, se muestra el listado de mensajes analizados con el resultado obtenido del detector de discurso de odio como se muestra en la *Ilustración 4.11*.

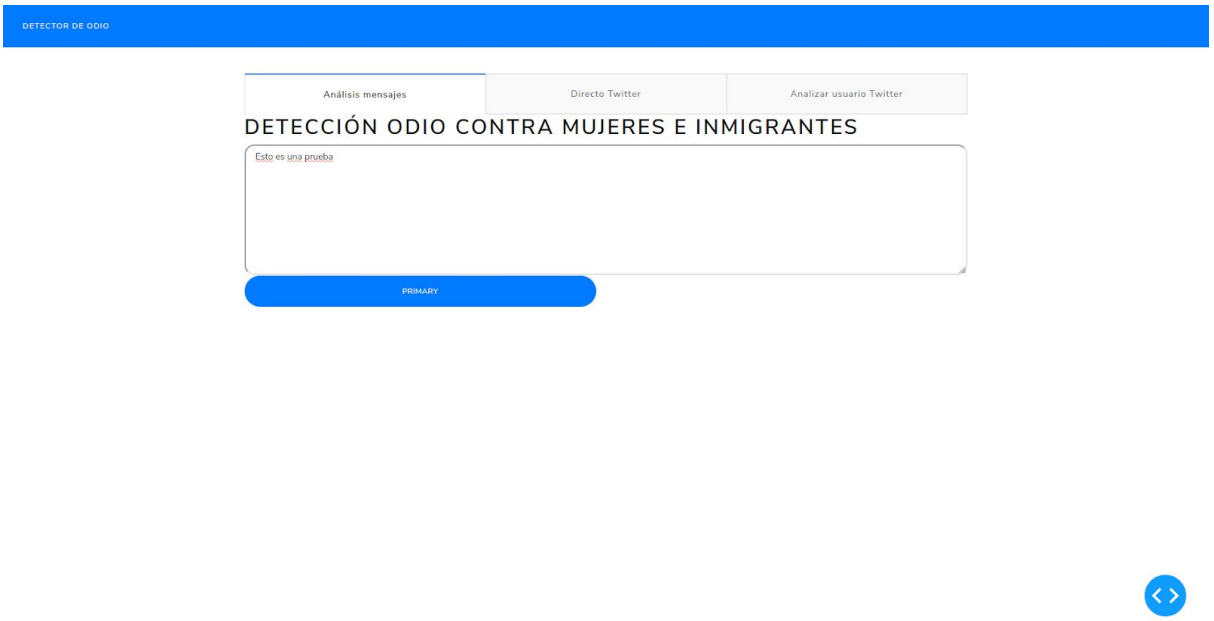

Ilustración 4.11. Estado inicial aplicación web

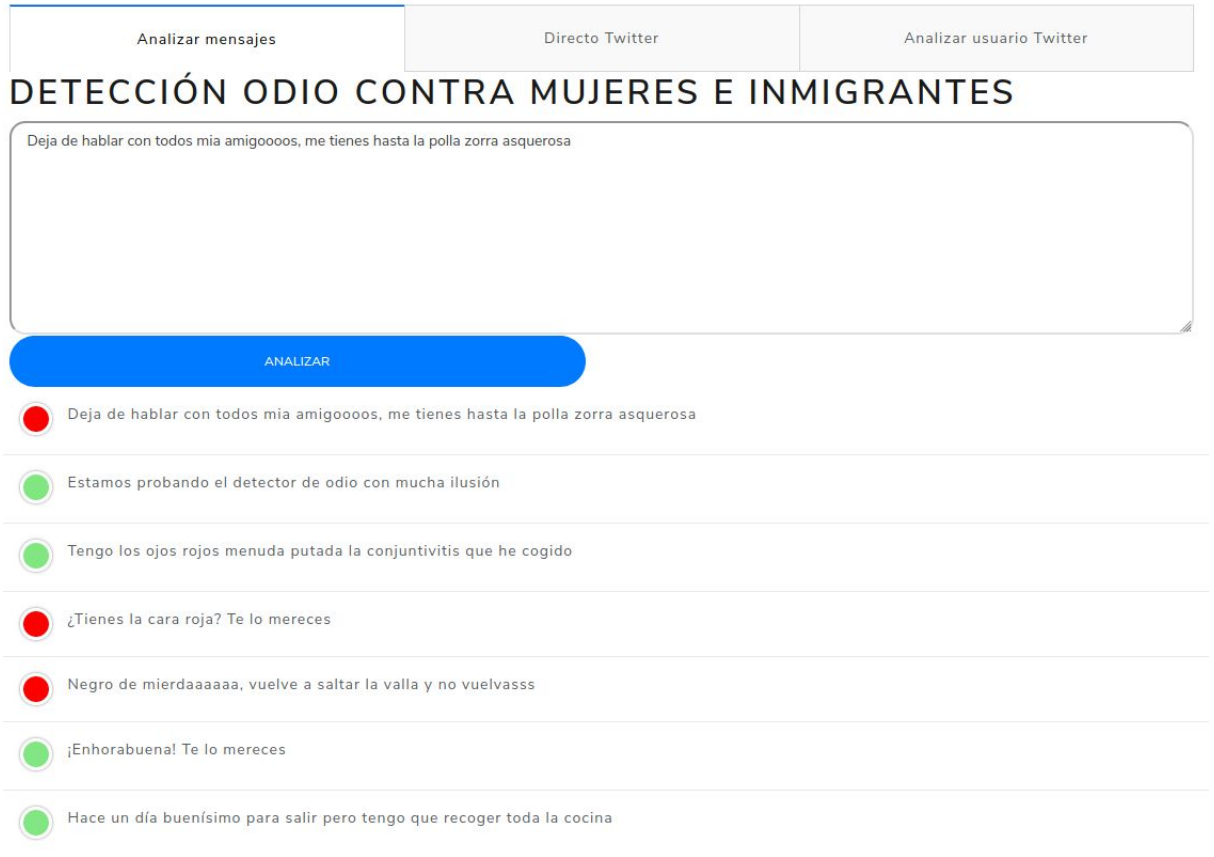

Ilustración 4.12. Ejemplo consulta de la aplicación web

Ilustración 4.13. Ejemplo consulta en directo

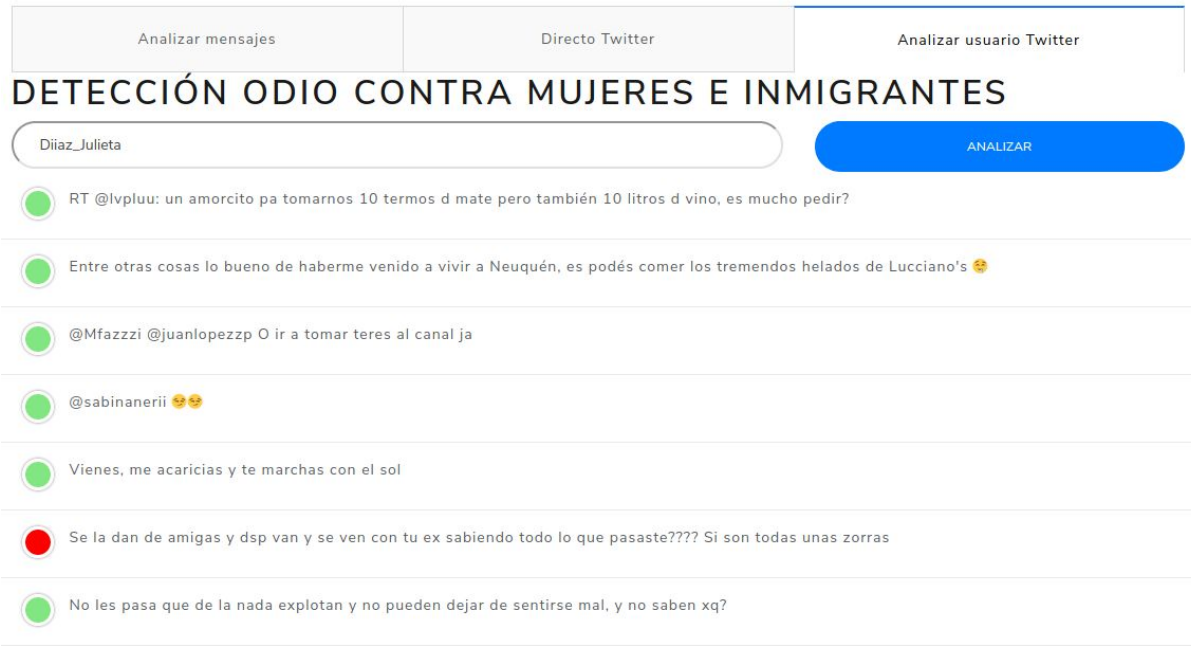

Ilustración 4.14. Ejemplo consulta usuario

4.7.5. Iteración 5

Iteración 5: "Como usuario quiero poder interactuar de diferentes formas con el monitor de discurso de odio"

La iteración 5 se corresponde a la historia de usuario "*Tipos interacción con monitor*" mostrada en la *Tabla 4.6.*

Uno de los principales objetivos de la aplicación web es permitir al usuario interactuar de diferentes formas con el monitor de discurso de odio. Para ello, se ha proporcionado la posibilidad de que el usuario introduzca cualquier mensaje de forma manual. Además, se han proporcionado herramientas para extraer y analizar mensajes de Twitter. Las diferentes secciones son:

- "Analizar mensajes": permite introducir un mensaje manualmente y el sistema devuelve si detecta discurso de odio.

- "Directo Twitter": permite realizar una búsqueda en directo en Twitter. La aplicación descarga contenido en *streaming* de Twitter, la analiza con el sistema de detección de odio y muestra todos los mensajes indicando si incluye discurso de odio.
- "Analizar usuario Twitter": permite consultar los últimos *tweets* de una cuenta de Twitter. Se muestra un listado de mensajes con su resultado del detector de discurso de odio.

Para la extracción de los *tweets* se ha utilizado la biblioteca *Tweepy* de *Python* que permite conectarse a la API de Twitter de una manera sencilla. Twitter da soporte a dos APIs distintas: Rest API y Streaming API. La primera funciona mediante consultas al servidor de Twitter y recibe una respuesta en formato JSON. La Streaming API proporciona tweets extraídos en tiempo real.

Para la consulta de *tweets* en directo Tweepy se encargará de comunicarse con la Streaming API mediante la clase StreamListener. Para ello hay que realizar un proceso de autentificación usando los tokens correspondientes proporcionados por Twitter Developer[18]. Una vez autenticado, se inicia una instancia *listener* con la consulta a enviar. La API de Twitter comienza a extraer *tweets* que se irán almacenando en una colección de la base de datos MongoDB. En paralelo, hay otro hilo que estará escuchando si hay tweets nuevos en la base de datos para analizarlos mediante el detector de odio y que se muestre en la aplicación web.

4.8. Herramientas para desarrollo del proyecto

Para el desarrollo del proyecto, se han necesitado utilizar herramientas para su desarrollo tanto tecnologías para el desarrollo del sistema de detección de discurso de odio como para la gestión del proyecto.

[18] https://developer.twitter.com/

4.8.1. Lenguajes de programación

4.8.1.1. Python

Python es un lenguaje de programación interpretado, esto significa que no es necesario compilar el código para poder ejecutarlo. Se trata de un lenguaje de programación multiparadigma, de programación orientativa y orientado a objetos. En los últimos años se ha hecho muy popular por las siguientes razones:

- Existen multitud de librerías, tipos de datos y funciones.
- Sencillez y velocidad en el desarrollo de código
- Multiplataforma
- Gratuito

4.8.1.2. JavaScript

JavaScript es un lenguaje de alto nivel, dinámico e interpretado, dialecto del estándar ECMAScript. Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo y débilmente tipado. JavaScript generalmente se utiliza como lenguaje para desarrollar funcionalidades web en el cliente.

4.8.1.3. HTML

HTML, siglas en inglés de *HyperText Markup Language*, se trata de un lenguaje de marcado para el desarrollo de páginas web. Es un estándar gestionado por World Wide Web Consortium (W3C)[19]. HTML es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones. Se desarrolla una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementandose con otros objetos como imágenes, vídeos, etc. Es el estándar que se ha tomado para la visualización de páginas web y todos los navegadores soportan.

[19] https://www.w3c.es/

4.8.2. Gestor de base de datos

4.8.2.1. MongoDB

MongoDB es un sistema sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++. MongoDB no almacena los datos en tablas, como lo hacen los gestores de base de datos más conocidos, sino que lo almacena en documentos estructurados en formato BSON similar a los JSON. Las características principales son:

- Consultas ad hoc. Permite hacer búsquedas por campos, consultas de rangos y expresiones regulares. Además, estas consultas pueden devolver un campo específico del documento.
- Indexación. La indexación es similar a las bases de datos relacionales. Cualquier campo puede ser indexado y añadir múltiples secundarios.
- Replicación. Soporta el tipo de replicación primario-secundario.
- Balanceo de carga. MongoDB tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balanceo de carga o replicación de datos.
- Almacenamiento de archivos. Puede ser utilizado también como un sistema de ficheros.

4.8.3. Framework

4.8.3.1. Dash

Dash es un framework de Python para la construcción de aplicaciones web para analítica. Dash está escrito sobre Flask, Plotly.js y React.js. Dash es ideal para construir aplicaciones de visualización de datos con interfaces personalizables para el usuario.

Las aplicaciones de Dash se renderizan en el navegador web. Como las aplicaciones se ven en navegadores web, es multiplataforma y permite visualizarlo en móviles.

Dash es una biblioteca de código abierto, liberada bajo la licencia permisiva del MIT. Plotly desarrolla Dash y ofrece una plataforma para gestionar las aplicaciones de Dash en un entorno empresarial.

4.8.3.2. Bootstrap

Bootstrap[20] es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con posibilidad de modificar. Las ventajas de utilizar Bootstrap son:

- Es de código abierto.
- Mantenido y actualizado por Twitter.
- Compatible con la mayoría de los navegadores web.
- Gran cantidad de documentación.
- Incluye Grid system.
- Plantillas de adaptación responsive.
- Integrado librerías JavaScript.

4.8.4. Gestión de proyectos

4.8.4.1. Git

Git es un software de control de versiones desarrollado por Linus Torvalds. Su propósito es llevar registro de los cambios de los archivos del directorio y coordinar el trabajo que varias personas realizan paralelamente. Las características más destacadas son:

- Software libre.
- Permite tener historial completo de versiones.

[20] https://getbootstrap.com/

- Sistema de trabajo con ramas.
- No depende de un repositorio central.

4.8.4.2. Trello

Trello es una aplicación basada en el método Kanban. Sirve para gestionar tareas permitiendo el trabajo en grupo de forma colaborativa mediante tableros virtuales.

Es muy útil para la gestión de proyectos ya que se pueden representar distintos estados y compartirlos con diferentes personas que formen el proyecto. Sus principales características son:

- La versión gratuita es útil y con amplio abanico de funcionalidades.
- La aplicación online es compartida en tiempo real.
- La aplicación es muy simple e intuitiva.
- Tiene sistema de notificaciones. Las notificaciones pueden avisar a través de correo electrónico.
- Motor de búsqueda para tareas.

5. CONCLUSIONES

Durante este capítulo, se exponen mis valoraciones personales sobre el proyecto y los futuros trabajos a desarrollar sobre esta aplicación para lograr un mejor funcionamiento.

5.1. Valoración personal

Llegado a este punto, doy por finalizado el proyecto ya que se han conseguido alcanzar los objetivos que se habían marcado al inicio del trabajo fin de máster.

El desarrollo de este proyecto comienza con la idea de continuar ampliando mis conocimientos en el área del PLN. En el cuarto curso del Grado de Ingeniería Informática cursé la asignatura Procesamiento del Lenguaje Natural, donde comencé a conocer este campo de la informática que me apasionaba. Además, logré en paralelo una beca Ícaro con el grupo de Sistemas Inteligentes de Acceso a la Información (SINAI) de la Universidad de Jaén, que trabaja en el ámbito del PLN. Mi Trabajo Fin de Grado (TFG) se centraba en esta área, desarrollando un sistema de reconocimiento de emociones en español, debido a la escasez que había en este campo, tutorizado por Mª Teresa y Salud María.

La satisfacción obtenida con las experiencias en el PLN, me llevó a querer seguir profundizando en esta área. avanzando un paso más hacia la detección del discurso de odio en las redes sociales en mujeres e inmigrantes, que permite ayudar a solucionar un problema actual de la sociedad como es el odio hacia estos grupos sociales. También me ha permitido simular la participación en una de las competiciones de esta área para poder analizar y comparar el resultado de mi sistema, cosa que no pude hacer en mi TFG.

Por último, me gustaría agradecer a mis tutoras Mª Teresa y Flor Miriam la confianza y el tiempo depositado en mí. Permitiendo mejorar tanto en el ámbito profesional como personal.

5.2. Conclusiones

Existen una serie de mejoras posibles a incluir en el proyecto. A continuación, expongo algunas de las mejoras:

- Entrenar el sistema para otro tipo de textos en redes sociales. En la actualidad, el detector de discurso de odio ha sido entrenado para trabajar con mensajes cortos y con las características de los *tweets*.
- Generar un corpus etiquetado con mensajes de mayor extensión que los *tweets*. Esto permitiría evaluar si nuestro sistema sólo funciona con *tweets* o también con otro tipo de textos, con diferente extensión o forma de expresión. También permitiría desarrollar un sistema especializado en textos más largos.
- Incorporar redes neuronales al clasificador de discurso de odio para conseguir mejores resultados. Debido a su constitución las redes neuronales son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc.
- Generar un sistema detector de odio multilingüe. Habría que realizar un estudio de las diferentes lenguas para ver las características similares del discurso de odio en todos los idiomas. Para extraer características se podría apoyar en WordNet que es una base de datos léxica multilingüe.
- Añadir confianza del detector de discurso de odio. El sistema no solo muestre el resultado del clasificador, sino la confianza que tiene en el resultado proporcionado.
- Monitorizar Twitter. Se podría incorporar que el sistema esté continuamente escuchando en Twitter detectando los tweets con discurso de odio y reportarlos a esta red social.
- Añadir a la aplicación web una sección que muestre mensajes para que lo etiqueten los usuarios. Esto permitirá tener un corpus más extenso y etiquetado.

ANEXO A: MANUAL DE INSTALACIÓN

A.1. Python y sus librerías

Para el desarrollo del proyecto, se ha utilizado como lenguaje principal de programación Python. La versión utilizada ha sido la 3.6.9. Si se va a utilizar un sistema operativo, se debe consultar el repositorio oficial[21] para instalaciones. En este manual se va a seguir una instalación para el sistema operativo Ubuntu. Para instalar Python hay que ejecutar el siguiente comando:

```
sudo apt-get install python3.6.9.
```

Para comprobar si se ha instalado correctamente la versión indicada, se va a consultar la versión Python de nuestro sistema operativo:

```
python3 -v
```

Para la instalación de las librerías se va a utilizar la herramienta *pip*. Se ejecuta el comando:

```
sudo apt-get install python3-pip
```

Para instalar todas las librerías dependientes del proyecto y su versión necesaria se ha creado un fichero llamado *requirements.txt.* Para instalar las librerías se ejecuta:

```
pip3 install -r requirements.txt
```

Cuando se haya completado la instalación de las librerías necesarias, se siguiente pasar a la instalación de la base de datos (A.2).

[21] https://www.python.org/downloads/

A.2. Bases de datos

Como en el *Anexo A.1.* se detalla la instalación en el sistema operativo Ubuntu. Para instalar MongoDB se ejecuta:

```
sudo apt-get install -y mongodb-org
```

MongoDB se instala como un servicio. a continuación se inicia el servicio con:

```
sudo systemctl start mongodb
```

Para comprobar si se ha activado el servicio, se comprueba que su estado sea "*active*", lanzando el comando:

```
sudo systemctl status mongodb
```

ANEXO B. MANUAL DE USUARIO

En este anexo, se muestra a modo de visita guiada, un manual de la aplicación web para resolver cualquier tipo de duda.

En primer lugar, hay que iniciar el servidor con la aplicación web. Para ello, hay que situarse en el directorio de la aplicación y ejecutar el siguiente comando:

```
python3 visualizacion.py
```

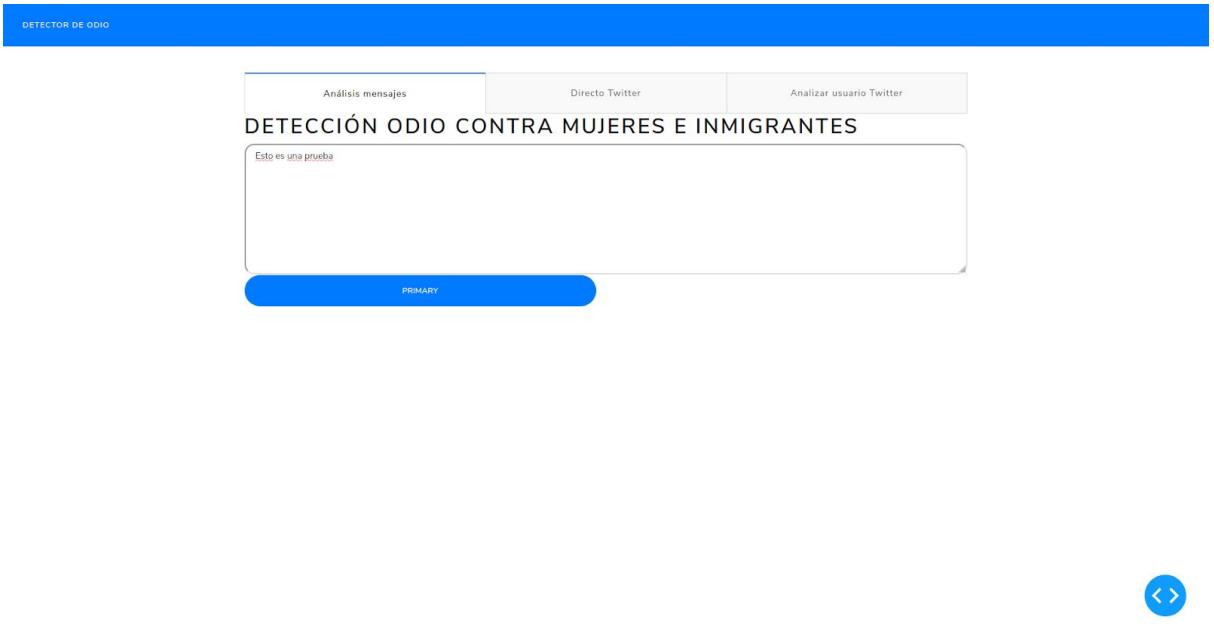

Ilustración A.1. Aplicación web "Analizar mensajes"

En el estado actual, el usuario puede escribir cualquier tipo de mensaje para analizar. Una vez se ha introducido el mensaje, al pulsar el botón "Analizar" la aplicación mostrará en el listado de mensajes el nuevo texto con el resultado del clasificador.

Si se selecciona el botón "Directo Twitter", aparecerá la pantalla de la *Ilustración A.2.* En la caja de texto se introduce la búsqueda a realizar en Twitter. Al

pulsar en el botón “Analizar” la aplicación comienza a recolectar mensajes que contengan la búsqueda realizada y los muestra en la pantalla.

Ilustración A.2. Aplicación web “Directo Twitter”

Si se selecciona la sección “Analizar usuario Twitter” se mostrará una pantalla similar a *Ilustración A.3*, en el cuadro de texto se introduce el usuario de Twitter a analizar sin el @, es decir si se quiere consultar el usuario @Pontifex_es hay que buscar por “Pontifex_es”.

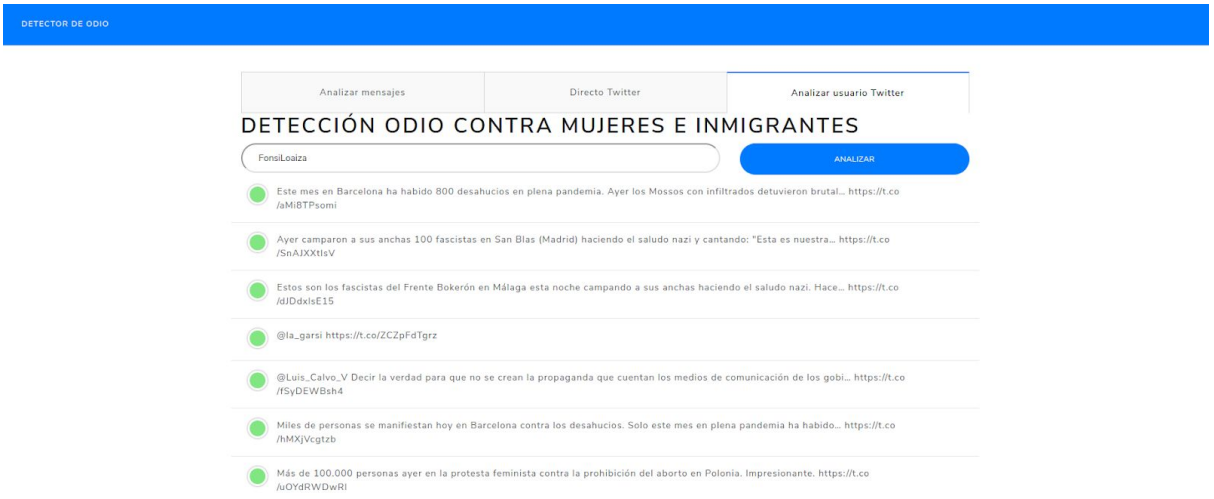

Ilustración A.3. Aplicación web “Analizar usuario Twitter”

ANEXO C: ÍNDICE DE ILUSTRACIONES

ANEXO D. ÍNDICE DE TABLAS

BIBLIOGRAFÍA

[1] Waseem, Z., Chung, W. H. K., Hovy, D., & Tetreault, J. (2017, August). Proceedings of the First Workshop on Abusive Language Online. In *Proceedings of the First Workshop on Abusive Language Online*.

[2] Kumar, R., Ojha, A. K., Zampieri, M., & Malmasi, S. (2018, August). Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*.

[3] Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., & Klenner, M. (2019). Overview of GermEval Task 2, 2019 shared task on the identification of offensive language.

[4] Basile, V., Bosco, C., Fersini, E., Debora, N., Patti, V., Pardo, F. M. R., ... & Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation* (pp. 54-63). Association for Computational Linguistics.

[5] Plaza-Del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2020). Detecting misogyny and xenophobia in Spanish tweets using language technologies. *ACM Transactions on Internet Technology (TOIT)*, *20*(2), 1-19.

[6] Pérez, J. M., & Luque, F. M. (2019, June). Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 64-69).

[7] Vega, L. E. A., Reyes-Magaña, J. C., Gómez-Adorno, H., & Bel-Enguix, G. (2019, June). MineriaUNAM at SemEval-2019 Task 5: Detecting hate speech in Twitter

using multiple features in a combinatorial framework. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 447-452).

[8] Gómez-Adorno, H., Enguix, G. B., Sierra, G., Sánchez, O., & Quezada, D. (2018). A Machine Learning Approach for Detecting Aggressive Tweets in Spanish. In *IberEval@ SEPLN* (pp. 102-107).

[9] Tellez, E. S., Moctezuma, D., Miranda-Jiménez, S., & Graff, M. (2018). An automated text categorization framework based on hyperparameter optimization. *Knowledge-Based Systems*, *149*, 110-123.

[10] Fersini, E., Nozza, D., & Rosso, P. (2018). Overview of the evalita 2018 task on automatic misogyny identification (ami). *EVALITA Evaluation of NLP and Speech Tools for Italian*, *12*, 59.

[11] Bosco, C., Patti, V., Bogetti, M., Conoscenti, M., Ruffo, G. F., Schifanella, R., & Stranisci, M. (2017). Tools and resources for detecting hate and prejudice against immigrants in social media. In *SYMPOSIUM III. SOCIAL INTERACTIONS IN COMPLEX INTELLIGENT SYSTEMS (SICIS) at AISB 2017* (pp. 79-84). AISB.

[12] Fersini, E., Nozza, D., & Rosso, P. (2018). Overview of the evalita 2018 task on automatic misogyny identification (ami). *EVALITA Evaluation of NLP and Speech Tools for Italian*, *12*, 59.

[13] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).

[14] Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Sung, Y. H. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

[15] Aragón, M. E., Jarquín, H., Gómez, M. M. Y., Escalante, H. J., Villaseñor-Pineda, L., Gómez-Adorno, H., ... & Posadas-Durán, J. P. (2020, September). Overview of mex-a3t at iberlef 2020: Fake news and aggressiveness analysis in mexican spanish. In *Notebook Papers of 2nd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Malaga, Spain*.

[16] Molina-Gonzalez, M. D. (2016). Generación de recursos para análisis de opiniones en español.

[17] Rangel, I. D., Sidorov, G., & Guerra, S. S. (2014). Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. Onomazein, (29), 31-46.