

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

**Diseño y creación de un sistema identificador
de violencia doméstica psicológica por voz**

Robert Jarod Meza Garcia

Para optar el Título Profesional de
Ingeniero de Sistemas e Informática

Huancayo, 2021

Repositorio Institucional Continental
Tesis digital



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

AGRADECIMIENTOS

Quiero expresar mis agradecimientos al ingeniero Job Daniel Gamarra Moreno, por su aporte y participación como asesor en el desarrollo de esta tesis. Debo destacar, por encima de todo, su disponibilidad y paciencia que hizo que la tesis tuviera beneficios a nivel científico y personal. No cabe duda de que su participación ha enriquecido el trabajo realizado y, además ha significado el surgimiento de un sólido proyecto.

A mi papá, Roberto Edwin Meza Segura que fue de mucho apoyo en la recolección de información, y a mi madre Marilú Jacqueline Garcia Berrocal por el ánimo de seguir elaborando mi estudio tratado en este informe.

A la asesoría virtual de los portales de educación por internet, que me dieron una vista completa de las tecnologías y el uso que se les puede dar para generar ideas innovadoras y hacerme dar cuenta que toda constancia es recompensada.

DEDICATORIA

A mis padres, por enseñarme y hacerme la persona que soy, por darme a entender que nada es irresoluble solo la mortalidad humana, por confiar en mi capacidad y mostrarme a mí mismo que sí se puede.

ÍNDICE

Agradecimientos	ii
Dedicatoria.....	iii
Índice	iv
Índice De Figuras.....	viii
Índice De Tablas	x
Resumen	xi
Abstract.....	xii
Introducción	xiii
CAPÍTULO I.....	15
PLANTEAMIENTO DEL ESTUDIO.....	15
1.1. Planteamiento del problema	15
1.2. Formulación del problema.....	16
1.2.1. Problema general	16
1.2.2. Problemas específicos	16
1.3. Objetivos	16
1.3.1. Objetivo general	16
1.3.2. Objetivos específicos.....	16
1.4. Justificación	17
1.4.1. Justificación práctica	17
1.4.2. Justificación metodológica.....	17
CAPÍTULO II.....	19
MARCO TEÓRICO	19
2.1. Antecedentes de la investigación	19
2.1.1. Antecedentes internacionales	19
2.1.2. Antecedentes nacionales	20
2.2. Bases teóricas	21
2.2.1. Procesamiento del lenguaje natural.....	21
2.2.2. Teorema de Bayes.....	22
2.2.3. Fases de entrenamiento	23
2.2.4. Matriz de confusión.....	23
2.2.5. Lexicografía de los insultos.....	25
2.2.5.1. Frases nominales	25

2.2.5.2. Frases adjetivales	25
2.2.5.3. Frases verbales	26
2.2.5.4. Frases prepositivas.....	26
2.2.6. PMBOK.....	27
2.2.7. Scrum.....	28
2.3. Definición de términos básicos.....	31
2.3.1. Violencia psicológica.....	31
2.3.2. Sintagma o frase	31
2.3.3. Frases nominales.....	31
2.3.4. Frases adjetivales	31
2.3.5. Frases verbales	32
2.3.6. Frases prepositivas.....	32
2.3.7. Teorema de Naïve Bayes	33
2.3.8. Razonamiento inductivo.....	33
CAPÍTULO III.....	34
METODOLOGÍA.....	34
3.1. Metodología aplicada para el desarrollo de la solución	34
3.1.1. Metodología aplicada a la gestión del proyecto	34
3.1.2. Metodología aplicada a la solución de la inteligencia artificial	36
3.1.2.1. Población.....	36
3.1.2.2. Muestra.....	37
3.1.2.3. Instrumento de validación de contenido.....	37
3.1.2.4. Indicadores	38
3.1.3. Metodología para la recolección de datos.....	39
3.2. Metodología del procesamiento del lenguaje natural	41
3.2.1. Nivel morfológico	41
3.2.2. Nivel sintáctico	41
3.2.3. Nivel semántico.....	43
CAPÍTULO IV	45
DISEÑO	45
4.1. Designación de roles	45
4.2. Objetivos de la organización	45
4.3. Definición del alcance.....	46
4.3.1. Alcance del producto	46

4.3.2. Restricciones del producto.....	46
4.3.3. Alcance del proyecto.....	47
4.4. Definición del EDT.....	47
4.5. Identificación de requerimientos.....	49
4.6. Pila de producto.....	50
4.7. Planificación de <i>Sprints</i>	57
4.8. Análisis de la solución.....	58
4.8.1. Propuesta.....	58
4.8.2. Arquitectura de la solución.....	58
4.8.2.1. Lado del cliente.....	59
4.8.2.2. Google cloud, service “Speech to text”.....	59
4.8.2.3. Servicio del módulo reconocedor de frases.....	60
4.8.2.4. Servicios <i>Rest</i> para consumo de recursos de servidor.....	60
4.8.2.5. Cliente web.....	60
4.8.3. Diseño de interfaz gráfica.....	60
4.8.4. Análisis técnico y económico.....	63
CAPÍTULO V.....	66
CONSTRUCCIÓN.....	66
5.1. Construcción.....	66
5.1.1. Lado servidor.....	66
5.1.1.1. Servidor en <i>Python</i>	67
5.1.1.2. Servidor en <i>Typescript</i>	69
5.1.2. Lado cliente.....	72
5.1.2.1. Aplicación móvil.....	72
5.1.2.2. Aplicación web (lado administrativo).....	75
5.2. Pruebas.....	78
5.2.1. Pruebas del modelo clasificador.....	78
5.2.2. Pruebas del aplicativo.....	79
5.3. Matriz de confusión, medición de eficacia.....	81
5.3.1. Exactitud de clasificación.....	82
5.3.2. Tasa de error de clasificación.....	82
5.3.3. Sensibilidad de clasificación.....	82
5.3.4. Especificidad de clasificación.....	82
5.3. Discusión de resultados.....	83

Conclusiones.....	85
Trabajos Futuros.....	87
Lista De Referencias	88
Anexos	91

ÍNDICE DE FIGURAS

Figura 1. Nomenclatura de las variables del teorema de Bayes. Adaptado de Algoritmos Naive Bayes: Fundamentos e Implementación	23
Figura 2. Estructura del sintagma preposicional.....	32
Figura 3. Diseño y flujo del procesamiento del lenguaje natural del sistema ...	41
Figura 4. Proceso de sistema de etiquetado	42
Figura 5. Diseño del análisis pragmático.....	43
Figura 6. EDT del proyecto.....	48
Figura 7. Arquitectura de la solución propuesta	59
Figura 8. Diagrama de flujo de interfaces de la aplicación	62
Figura 9. Demostración del envío de información mediante el método GET de http	67
Figura 10. Código del servidor web, consumo de servicios creados.....	68
Figura 11. Conexión con la base de datos MongoDB.....	68
Figura 12. Código de entrenamiento, primera parte.....	69
Figura 13. Código de servidor, módulo principal	70
Figura 14. Modelo del documento de usuario	71
Figura 15. Estrategia de autenticación a través de Json Web Token.....	72
Figura 16. Vista del prototipo móvil en desarrollo.....	73
Figura 17. Arquitectura Bloc para la aplicación Painless.....	74
Figura 18. Representación gráfica del patrón Bloc.....	75
Figura 19. Arquitectura de carpetas para la creación de componentes	76
Figura 20. Pantalla administrativa, visualización de cantidad de frases detectadas en el tiempo	77
Figura 21. Pantalla de ubicaciones de frases ofensivas clasificadas	77
Figura 22. Pantalla de administración de usuarios maestros	78
Figura 23. Modal de agregar nuevo usuario.....	78
Figura 24. Resultados en escala de usabilidad realizadas en los formularios .	80
Figura 25. Tiempo de respuesta de la aplicación frente a la clasificación.....	80
Figura 26. Muestra estadística del uso y detección de frases ubicadas en el tiempo	81

Figura 27. Muestra de las últimas alertas registradas dentro de las 48 horas por
cuenta de correo electrónico 81

ÍNDICE DE TABLAS

Tabla 1. Matriz de confusión	24
Tabla 2. Cuadro de insultos de clase nominal.....	25
Tabla 3. Cuadro de frases adjetivales	26
Tabla 4. Cuadro de frases verbales	26
Tabla 5. Cuadro de frases prepositivas.....	27
Tabla 6. Áreas de conocimiento, gestión del proyecto mediante PMBOK	34
Tabla 7. Estructura de tareas según Scrum	35
Tabla 8. Indicadores en matriz de confusión.....	39
Tabla 9. Frases consideradas en el instrumento de validación.....	40
Tabla 10. Asignación de roles	45
Tabla 11. Pila de requerimientos.....	49
Tabla 12. Pila de producto del Sprint 1	51
Tabla 13. Pila de producto del Sprint 2	53
Tabla 14. Pila de producto Sprint 3	54
Tabla 15. Pila de producto Sprint 4	55
Tabla 16. Pila de producto Sprint 5	56
Tabla 17. Planificación de historias de las iteraciones	57
Tabla 18. Número de violencia registradas en el año 2019	61
Tabla 19. Tabla de costo directo unitario del proyecto	63
Tabla 20. Tabla de costos indirectos del proyecto	63
Tabla 21. Tabla de costos totales por el proyecto	64
Tabla 22. Cronograma de actividades en base a iteraciones.....	64
Tabla 23. Pruebas de frases agresivas	79
Tabla 24. Tabla de pruebas de frases no agresivas.....	79
Tabla 25. Matriz de confusión de resultados de datos de prueba	81

RESUMEN

En la presente tesis “*Diseño y creación de un sistema identificador de violencia doméstica psicológica por voz*”, está dedicada a resolver el problema de no poder contar con instrumentos de acción que permitan identificar violencia psicológica por voz durante la incidencia de violencia, por lo que el objetivo de diseñar y crear una aplicación capaz de clasificar violencia doméstica psicológica por voz, grabar el incidente y enviar la ubicación de la clasificación agresiva, se ha desarrollado como una aplicación híbrida hecha con *Flutter*, un SDK capaz de renderizar componentes nativos tanto para Android, como para Iphone. Para el desarrollo de la aplicación se trabajó bajo el método *Scrum*, siendo una tecnología usada frecuentemente para el desarrollo de aplicación bajo una metodología ágil y resultados incrementales dentro de ciclos cortos de entrega. Para la recopilación de datos que son parte del entrenamiento fueron a través del inventario fraseológico de los insultos, siendo validado a través de un instrumento de validación llenado por personas coterráneas habitantes y nativos del lenguaje español. El sistema se enfocó en la clasificación de frases agresivas, identificando la estructura de las frases a un nivel morfosintáctico, a través de “multigramas” y palabras frecuentes dentro de la muestra trabajada. Posteriormente, este sistema ha sido implementado en un servidor HTTP y una aplicación móvil que permite el uso del servicio. Finalmente, dieron resultados suficientes para asegurar la validez de la empleabilidad de las frases. Se llegó a las conclusiones, que el sistema cumple con principios de usabilidad y así como también de funcionalidad dando un 95.24% de eficacia y un índice de usabilidad de 88.5%.

Palabras claves: *machine learning*, procesamiento de lenguaje natural, sistema inteligente, violencia familiar.

ABSTRACT

This thesis entitled “Design and creation of a system to identify psychological domestic violence by voice” it is dedicated f to the problem of not being able to have action instruments that allow identifying psychological violence by voice during the incidence of violence, so the objective of designing and creating an application capable of classifying psychological domestic violence by voice, recording the incident and sending the location of the aggressive classification, developed as a hybrid application made with Flutter, an SDK capable of rendering native components for both Android and Iphone. For the development of the application, we worked under the Scrum method, being a frequently used technology for application development under an agile methodology and incremental results within short delivery cycles. For the collection of data that are part of the training, they were through the phraseological inventory of the insults, being validated through a validation instrument filled out by compatriots’ inhabitants of and natives of the Spanish language. The system focused on the classification of aggressive phrases, identifying the structure of the phrases at a morphosyntactic level, through “multigrams” and frequent words within the sample worked. Subsequently, this system has been implemented in an HTTP server and a mobile application that allows the use of the service. Finally, they gave sufficient results to ensure the validity of the employability of the sentences. The conclusions were reached that the system complies with the principles of usability and functionality, giving 95.24% efficiency and a usability index of 88.5%.

Keywords: domestic violence, machine learning, natural language processing, smart system.

INTRODUCCIÓN

El presente trabajo se enfoca en un tema específico de la violencia familiar, la violencia psicológica, siendo este un problema que ha sido registrado en un número de casos cada año en crecimiento hasta la actualidad, donde la tecnología y métodos para la solución de problemas analíticos y de clasificación han podido ser resueltos por computadoras, sistemas inteligentes capaces de discernir y analizar cantidades de datos para poder generar un algoritmo lo suficientemente funcional para un propósito en específico, es por ello que como objetivo de este trabajo es el diseño y creación de un sistema capaz de reconocer frases agresivas o violentas dentro de un espacio delimitado como lo es El Tambo, Huancayo, en el departamento de Junín, en el año 2021; recolectando información oportuna para el diseño y construcción, analizar y encontrar características lingüísticas de las frases violentas para el entrenamiento y, por último, la construcción de este sistema inteligente a través de procesamiento de lenguaje natural. Es por lo que este trabajo está estructurado así:

En el primer capítulo se planteó el problema, enunciando el problema social que es la violencia familiar y su incremento a través del tiempo reportado en los Centros de Emergencia Mujer instalados por el Ministerio de la Mujer y Poblaciones Vulnerables (MIMP), denotando que esta organización cuenta solo con medidas de prevención, concientización y atención después de los incidentes de violencia, principalmente agresiones verbales.

En el segundo capítulo se desarrolló el marco teórico, muestra los antecedentes del uso de inteligencia artificial y procesamiento de lenguaje natural para el análisis de sentimientos, procesados en texto y dando resultados, clasificando a través de algoritmos diversos, ya sean Teorema de Bayes, distancia de vectores entre palabras, uso de estadística y probabilidades, dando como resultado clasificaciones con resultados aceptables; luego se da a conocer la teoría de metodología ágil Scrum, y la guía para administración de proyectos PMBOK, del mismo modo los principios de diseño enfocado a la experiencia de

usuario y la forma de construcción del software en las distintas tecnologías empleadas.

En el tercer capítulo se desarrolló la metodología, detallando las características y etapas de la metodología de la gestión del proyecto, usando herramientas de la guía de PMBOK y juntándolas con la metodología ágil *Scrum*, así como también la metodología para la construcción del modelo de procesamiento natural y la metodología para la recolección de datos.

En el cuarto capítulo se desarrolló el análisis y diseño de la solución, se muestran los procedimientos y requerimientos documentados en el proceso del desarrollo del proyecto, haciendo consideraciones de estados determinados.

En el quinto capítulo se desarrolló la construcción y pruebas, muestra el desarrollo de la aplicación y los sistemas a los que recurre, los cuales se construyen bajo guías y buenas prácticas de prácticas de programación, como son las arquitecturas de los archivos y métodos de programación, a la vez demuestra el resultado de las clasificaciones en el ambiente de pruebas para clasificación de frases ofensivas, concluyendo con una exactitud del 95.24%.

Finalmente, se encuentran las conclusiones, trabajos futuros, lista de referencias y anexos.

CAPÍTULO I

PLANTEAMIENTO DEL ESTUDIO

1.1. Planteamiento del problema

“Un número de casos de víctimas de feminicidios [...] fueron registrados desde enero del 2009 hasta diciembre del 2019, con un total de 1 318 de casos de feminicidios y 2016 casos de tentativa, en total 3334 casos de violencia feminicida” (1). Se evidencia que esta problemática a través de los años ha sido constante e incluso en aumento por periodos, como lo fue el año 2018 con 304 feminicidios, y en el 2019 con 404, este aumento en cifras refleja un problema en la sociedad, dándose el 2 de junio de 2018, “un comunicado del presidente de la República Ing. Martin Vizcarra, [...] que declaró de interés y prioridad nacional del Estado la lucha contra la violencia hacia las mujeres” (2).

En el año 2019 se atendieron 181 885 casos de violencia familiar, se atendieron en todo el año a nivel nacional, a comparación del año 2018 que fueron 133 697 casos (1).

La consecuencia de estos sucesos dentro de las familias es que no solo se ve afectada la víctima, sino también los hijos dentro de este hogar, como lo muestra un estudio titulado “*Influencia de la familia sobre las conductas antisociales en adolescentes de Arequipa-Perú*” (3) en el que concluye que existen más varones adolescentes, en comparación con las mujeres que admiten

haber realizado conductas agresivas (golpes, insultos o actitudes sexuales) y la consecuencia en las mujeres teniendo conductas de impuntualidad.

A pesar de los planes de acción del 2019 del MIMP que definen medidas de predicción mediante lineamientos estratégicos que definen la implementación de cámaras en las calles, más centros de emergencias a la mujer, nuevos defensores públicos y más (1). Dejando la necesidad de tratar el acto o ser identificado durante el suceso, es así que esta tesis propone el diseño y desarrollo de un sistema identificador de violencia doméstica psicológica por reconocimiento de voz, usando inteligencia artificial para la clasificación de frases.

1.2. Formulación del problema

1.2.1. Problema general

¿Cómo identificar la violencia doméstica psicológica por voz al momento que el incidente esté en curso en el distrito de El Tambo, Huancayo, Junín en el periodo 2021?

1.2.2. Problemas específicos

- a) ¿Qué factores son pertinentes en un sistema que identifique las agresiones verbales?
- b) ¿Qué características lingüísticas resaltan en las agresiones verbales?
- c) ¿Cómo diseñar el modelo de procesamiento de lenguaje natural?
- d) ¿Cómo validar la eficiencia del modelo entrenado?

1.3. Objetivos

1.3.1. Objetivo general

Elaborar un sistema de reconocimiento de violencia doméstica psicológica por voz en el distrito de El Tambo, Huancayo, Junín en el periodo 2021.

1.3.2. Objetivos específicos

- a) Identificar los factores pertinentes en un sistema que identifique agresiones verbales.

- b) Identificar las características lingüísticas de la violencia psicológica familiar.
- c) Formular el diseño del sistema de procesamiento de lenguaje natural capaz de reconocer frases ofensivas o violentas.
- d) Medir la eficacia del modelo de procesamiento de lenguaje natural capaz de reconocer frases ofensivas y violentas.

1.4. Justificación

1.4.1. Justificación práctica

El Centro de Emergencia Mujer cuenta con distintos proyectos de concientización, prevención y atención de violencia doméstica, así como también el Ministerio de la Mujer y Poblaciones Vulnerables han llevado a cabo la ampliación y mejoramiento de servicios de prevención y promoción en sus centros, lo cual indica que ambas organizaciones están dispuestas a prevenir y atender incidentes ya ocurridos, mas no consiguen tener proyectos que puedan actuar al momento en que esté sucediendo el incidente de violencia, entonces, considerando el problema principal expuesto anteriormente, es un problema no contar con acciones o proyectos enfocados a socorrer o ayudar a posibles víctimas en el instante que se da la agresión; es más, el contener registros de un hecho tan cuestionable como lo es el afirmar la agresión verbal, es por eso que el desarrollo de la aplicación que detecte el instante y dé una alerta con los datos del usuario sería de gran ayuda.

De acuerdo con los problemas identificados, se sugiere que el uso de la aplicación del sistema de reconocimiento de agresiones verbales a través de la voz puede permitir al usuario contener un registro de grabaciones de las agresiones, así como la organización puede centrar la atención en las ubicaciones más incidentes en este tipo de agresiones verbales.

1.4.2. Justificación metodológica

Para el desarrollo de esta tesis se hará uso de un enfoque de desarrollo de una aplicación móvil multiplataforma con el SDK (*software*

development kit) de *Flutter*, ya que la herramienta *Flutter* es sumamente eficaz, aprovechando las funcionalidades y características del dispositivo, permitiendo dar el uso del hardware de este, como en este caso el uso y optimización del micrófono será indispensable (4), para el desarrollo e implementación del aplicativo, se usó la metodología *Scrum*, que permite procesos y cumplimiento de tareas en entornos complejos y los requisitos son variantes y prima la innovación (5).

Para que sea posible la identificación de agresiones es necesario realizarla mediante algún método de clasificación, por eso el uso de *machine learning*, una disciplina del ámbito de la inteligencia artificial se hará uso de teorema "*Bayes ingenuo*" (*Naïve Bayes*) como una forma para poder entrenar al sistema que sea capaz de clasificar las oraciones escuchadas, ya que el fin de la estadística bayesiana se basa en analizar los datos y tomar decisiones mediante ponderaciones como resultados de cada alternativa (6).

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes de la investigación

2.1.1. Antecedentes internacionales

“Airline sentiment visualization, consumer loyalty measurement and prediction using twitter data” (7), se centra en el análisis de los tweets relacionados con las aerolíneas basadas en su región: Europa, India, Australia y América para una predicción de consumo leal mediante el uso de inteligencia artificial, midiendo su nivel de satisfacción sobre el servicio de las empresas de sus regiones, en el análisis pudo dar una opinión general de los pasajeros hacia las aerolíneas teniendo percepción holística de sus clientes fidelizados, así como también clientes insatisfechos (7).

En el proyecto *“Sistema prototipo actuador por comandos de voz utilizando software libre”* presentan una aplicación informática que extrae la información de la voz del usuario, la cual se utiliza para administrar la activación/desactivación de un actuador periférico del computador personal. (8)

En el artículo *“Inventario fraseológico de las groserías en estudiantes de San Luis de Potosí México”* (9), se describen las formas lexicológicas sobre un inventario, dando una clasificación de las palabras

en nominales, adjetivales, esto permite al sistema la selección de agresiones dentro de algún diálogo dependiendo a la clase que sea (9).

En el trabajo *“Procesamiento de lenguaje natural para el análisis de lenguaje subjetivo”* (10), se analizó la forma de análisis de clasificación de opiniones de material textual enfocado a negocios, con el propósito de brindar ventajas competitivas frente al mercado que no prescindiera de la herramienta desarrollada apoyándose de un software existente llamado *“SentiWordNet”* (10).

2.1.2. Antecedentes nacionales

En la tesis *“Módulo de reconocimiento de voz a texto independiente de locutor para sistemas de diálogo”* (11), se determinan los pasos y la forma de cómo conseguir un sistema de procesamiento del habla continua, concluyendo que las etapas de un sistema de reconocimiento de voz son el análisis espectral de la señal de habla, comparación de la señal con modelos de unidades acústicas ya establecidos y búsqueda de significado de señal (11).

En años anteriores la tecnología de reconocimiento de voz ha sido usada para fines empresariales, satisfaciendo las necesidades y mejorando tiempos de espera, como se plasma en el trabajo *“Automatización del proceso de la toma de pedidos en el restaurante ‘Don Rulo SAC’, utilizando una aplicación móvil con reconocimiento de voz soportada por las API de Google”* (12), haciendo uso de la metodología *Iconix*, junto a las API de Google y el proyecto mediante programación por bloques de App Invento, se determina una forma de organizar una app con similares características de funcionalidad y propósito (12).

En el trabajo *“Diseño de un sistema inteligente utilizando procesamiento de lenguaje natural (PLN) para el reconocimiento de mensajes extorsivos”*, sacan provecho desde un enfoque social el uso de esta rama de inteligencia artificial para diseñar un sistema capaz de reconocer mensajes extorsivos, sosteniéndolos en su realidad y

manteniendo la sobriedad sobre las limitaciones que encontraron al momento de elaborar su trabajo (13).

2.2. Bases teóricas

2.2.1. Procesamiento del lenguaje natural

Como se explica:

“[...] El lenguaje se puede expresar de distintas maneras, como desde el punto de vista funcional lingüístico se define como una función que expresa pensamientos y comunicaciones entre la gente” (14).

El término de lenguaje natural se refiere tanto al lenguaje oral y escrito por el que se intenta transmitir un mensaje, y el hecho de comprender y componer mensajes son procesos muy complejos; sin embargo, algo que caracteriza al lenguaje natural es que se presenta de manera natural y sin mucho esfuerzo entre la gente y; por otro lado, la manera en cómo se comunican las computadoras requieren de un cálculo y protocolos matemáticos rigurosos (15).

Como formas de estudio se determinó que en el lenguaje natural existen diversos tipos de procesamientos que se reflejan claramente como una estructura en niveles, lo que permite separar aspectos de lo que trata cada nivel de proceso.

Niveles de procesamiento

- a. **Nivel fonológico:** consiste en la manera de cómo las palabras guardan relación con los sonidos que representan.
- b. **Nivel morfológico:** en este nivel se trata de analizar cómo es que se encuentra compuesta una palabra, el propósito de este nivel se centra en definir la forma, categoría o clase gramatical de cada palabra dentro de una oración. Por ejemplo, “si se busca interpretar la palabra “libros”, se obtiene que “*libr-*” es el lexema, que “-o-” y “-s” son morfemas dependientes flexivos de masculino y plural” (15).

- c. **Nivel sintáctico:** las palabras se unen para formar oraciones, siendo una estructura más compleja y con más significado, cada palabra se encarga de tener un rol dentro de esta nueva estructura brindando un significado (14).
- d. **Nivel semántico:** se ocupa de determinar el sentido de la oración basándose en la conexión y significado entre las palabras de las que está conformada, este nivel se limita a definir un sentido a las palabras con muchos significados, y así identificar el sentido en la representación semántica de la oración (15).
- e. **Nivel pragmático:** consiste en el uso de las oraciones y la manera en cómo varía su significado dependiendo de la situación (14).

2.2.2. Teorema de Bayes

Es uno de los métodos más usados en el proceso de *machine learning*, junto a los árboles de decisión y redes neuronales, es normalmente usada en tareas de clasificación de textos, sentimientos o filtros de mensajería en los correos electrónicos, la ventaja de este método es que no solo ofrece un análisis cualitativo, sino también cuantitativo, ya que este método da una medida probabilística de esas variables en el problema, dando así una probabilidad explícita de las hipótesis que se formulan (16).

Como se define en “*Análisis de clasificadores Bayesianos*” (17) *que*, por motivos de crear la capacidad de que el sistema mejore su eficiencia, se tomó interés, dedicación e investigación y desarrollo de esta área.

“Una red Bayesiana es un grafo acíclico dirigido en el que cada nodo representa una variable y cada arco una dependencia probabilística, en la cual se especifica la probabilidad condicional de cada variable dados sus padres” (17).

Ahora se representará el teorema de Bayes:

“[...]Teorema de Bayes (1764); sean A y B dos sucesos aleatorios cuyas probabilidades se denotan por $p(A)$ y $p(B)$ respectivamente, verificándose que $p(B) > 0$. Supongamos conocidas las probabilidades *a priori* de los sucesos A y B, es decir, $p(A)$ y $p(B)$; así como la probabilidad condicionada del suceso B dado el suceso A, es decir $p(B|A)$. La probabilidad *a posteriori* del suceso A conocido que se verifica el suceso B, es decir $p(A|B)$, puede calcularse a partir de la siguiente fórmula 1” (17):

$$p(A|B) = \frac{p(A, B)}{p(B)} = \frac{p(A)p(A|B)}{p(B)} = \frac{p(A)p(B|A)}{\sum_{A'} P(A')P(B|A')} \quad (1)$$

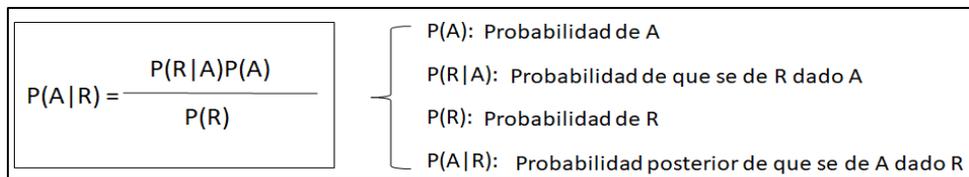


Figura 1. Nomenclatura de las variables del teorema de Bayes. Adaptado de Algoritmos Naive Bayes: Fundamentos e Implementación (18)

2.2.3. Fases de entrenamiento

- a. Preparar datos:** es necesario dar un tratamiento a los datos que serán introducidos al sistema, por lo que se requerirá de definir qué datos son los que entrarán.
- b. Entrenar al sistema:** el entrenamiento del sistema se concibe con la introducción de los datos correctos para una predicción precisa del sistema.
- c. Hacer pruebas al sistema:** luego del entrenamiento, se les somete a unas pruebas para que identifique la clasificación de cada frase introducida.

2.2.4. Matriz de confusión

Muy usada dentro del campo de la medicina para clasificación y diagnóstico de enfermedades “binarias”, quiere decir que todos los indicadores anteriormente puestos a prueba en un instrumento de clasificación dan un resultado positivo o negativo. Se le da uso para definir la eficacia del sistema o modelo de clasificación, es decir que un beneficio

de esta matriz es la de observar si el sistema no está fallando al dar resultados erróneos (19).

Tabla 1. Matriz de confusión

	Predicción	
	Positivos	Negativos
Observación Positivos	Verdaderos positivos (VP)	Falsos negativos (FN)
Negativos	Falsos positivos (FP)	Verdaderos negativos (VN)

Nota: para darle un buen uso al método de matriz de confusión, es necesario separar los datos de entrenamiento en dos “*datasets*” (conjunto de datos destinados a ser de utilidad de prueba o entrenamiento).

- Entrenamiento (80%)
- Prueba (20%)

La utilidad de dividir los datos de esta manera servirá para poder observar el comportamiento expuesto a data nueva (la data de prueba) (19).

Especificando la estructura de la tabla 1, se detalla lo siguiente:

- VP (verdaderos positivos) es el número de resultados positivos clasificados correctamente.
- VN (verdaderos negativos) es la cantidad de negativos que se calificaron correctamente como tal.
- FN (falsos negativos) es la cantidad de datos positivos calificados como negativos.
- FP (falsos positivos) es la cantidad de datos negativos calificados como positivos.

Exactitud o eficacia: permite responder a la pregunta:

¿Qué porcentaje de la data clasifica correctamente?

$$E = \frac{VN + VP}{Total\ de\ la\ muestra} \quad (2)$$

2.2.5. Lexicografía de los insultos

Para poder entender lo siguiente se debe entender que los insultos son compuestos de dos palabras (una frase), estas son de carácter morfológico, sintáctico o semántico, por lo que estas frases se clasifican en nominales, adjetivales, verbales y preposicionales (9).

2.2.5.1. Frases nominales

“[...]Las groserías en el lenguaje latino son estructuras fraseológicas altamente productivas y funcionales en la lengua hablada; estas permiten generar un sinnúmero de nuevas unidades para referirse al insulto” (9). Las frases nominales se caracterizan de tener como núcleo a un sustantivo seguido de más palabras enfocadas a modificar algo sobre él.

Tabla 2. Cuadro de insultos de clase nominal

Perra cachera	Cara de perra	Tetona de mierda	Hijo de tu puta madre
Hija de mierda	La mierda	Putra madre	Hija de tu perra madre
Hija de perra	Ni cagando	Cara de culo	Cholo mal parido
Hija de puta	Negro asqueroso	Cara de pinga	Pedazo de mierda
Hija de tu madre	Perra madre	Cabeza de miembro	Pedazo de cagada
Perro maldito	Perra malparida	Cara de vagina	Tu putísima madre
La concha de tu madre	Cabeza de pene	Cara de mierda	Tu reputa madre
Perra desgraciada	Cabeza de pinga	Jodida madre	Pinga de enano
Rata inmunda	Cara de caca	Cachera de mierda	Bueno para nada

Nota: adaptado de Inventario fraseológico de las groserías en estudiantes de San Luis Potosí (9)

2.2.5.2. Frases adjetivales

Las frases o sintagmas adjetivales son formados por una estructura con dos unidades morfológicas, una como núcleo siendo un adjetivo junto a una determinante, cualquiera que sea, ejemplo: “gorda desgraciada, adúltera estúpida”, “este tipo de frases tiene un inventario muy reducido en la lengua, por lo que su uso no es tan frecuente como la nominal o la verbal” (9). A continuación, se

presentan las frases adjetivales que aparecieron en la muestra en el siguiente cuadro:

Tabla 3. Cuadro de frases adjetivales

Maldito mocoso	Maldito bastardo	Cabro estúpido	Maricón pendejo
Maldita cabeza de mierda	Maldita culera	Ramera estúpida	Maricón imbécil
Maldita prostituta	Enferma perra	Facilona cojuda	Pinche perra
Maldito perro	Maldita prostituta	Perra sucia	Maldita golfa
Cabro estúpido	Maldito cabrón	Maldito estúpido	Maldita perra

Nota: adaptado de Inventario fraseológico de las groserías en estudiantes de San Luis Potosí (9)

2.2.5.3. Frases verbales

Otra estructura es la de verbo y un complemento. “Estas frases del tipo verbales son las más abundantes en el español, ejemplo: chingas a tu puta madre”; quiere decir, ve a molestar a tu mamá, normalmente una frase y su contexto es definida por una comunidad y el contexto en la que es usada podría variar su significado, a continuación, se verán unos ejemplos de sintagmas verbales (9).

Tabla 4. Cuadro de frases verbales

Agárrame esta	No seas maricón	Chúpamela	Vale pincho
Cállate el hocico	No seas perra	Mámamela	Vete a la mierda
Vete a la concha de tu madre	Rósamelo	Mama pinga	Vale pinga
Jálamela	No vales ni mierda	No jodas	Chupa mi cabeza
Jálatela	Te pasaste de imbécil	Vale mierda	No estés jodiendo

Nota: cuadro de frases verbales. Adaptado de Inventario fraseológico de las groserías en estudiantes de San Luis Potosí (9)

2.2.5.4. Frases prepositivas

Son aquellas frases que contienen preposiciones que refuercen al núcleo. Es decir que, a estas palabras les precede un adverbio, del cual le sucede un núcleo como un sustantivo. Las frases prepositivas recuperadas se muestran en la tabla 5.

Tabla 5. Cuadro de frases prepositivas

A la puta madre	Por la puta madre
A la mierda	Por la concha su madre

Nota: cuadro de frases verbales. Adaptado de Inventario fraseológico de las groserías en estudiantes de San Luis Potosí (9)

2.2.6. PMBOK

El PMBOK (*Project Management Body of Knowledge*), es una colección de procesos y áreas de conocimiento diseñadas y validadas, reconocidas dentro del estándar (IEEE Std 1490-2003) el cual provee como guía fundamentos de la gestión de proyectos, en esta oportunidad se dará el alcance de 8 áreas de conocimiento, para el uso paralelo a la metodología ágil *Scrum* (20).

- **Gestión de integración:** el área en donde se describen las necesidades, los objetivos y los entregables de forma general, incluye procesos y actividades que son importantes para identificar los procesos y actividades de dirección de los proyectos. En esta área de conocimiento se unifican las características de unificación, consolidación, comunicación y acciones integradoras (20).
- **Gestión de alcance:** es el área en donde se definen tanto los alcances del producto, como los alcances del proyecto, aquí se definen todos los procesos necesarios del producto. Necesarios para garantizar las tareas urgentes que estén identificadas y sean consideradas dentro del proceso. Se crea un plan de gestión de alcance que determina, documenta y gestiona las necesidades y los requisitos (20).
- **Gestión de tiempo:** la gestión del tiempo del proyecto contiene los procesos que se requieren para administrar el plazo dentro del tiempo del proyecto, la descripción general de los procesos para la planificación de la gestión de proyectos. La planificación de la gestión del cronograma es el proceso que determina las pautas, procedimientos y documentación para planificar, desarrollar, gestionar, ejecutar y controlar el cronograma del proyecto. El beneficio más importante de este proceso es que proporciona instrucciones a medida que se gestiona el plan del proyecto durante todo el período. La

definición de actividades es el proceso que identifica y las acciones de documentos (20).

- **Gestión de interesados:** es el proceso en donde se definen los interesados del proyecto y en donde se verán involucrados, siendo afectados en este, dependiendo el nivel de involucración del individuo (20).
- **Gestión de costos:** el área en la que se definen los costos, se estima, controla, presupuesta y financia, de forma que permite hacer una gestión de los costos del proyecto (20).
- **Gestión de calidad:** la gestión de la calidad del proyecto integra los procesos y la operación del negocio para establecer pautas sólidas, buenos objetivos y responsabilidades para que el proyecto satisfaga las necesidades en las que se mantiene una buena planificación de la gestión; es el proceso de identificación de requerimientos o requisitos, considerando buenas condiciones para el proyecto y lo que se puede tomar y documentación de cómo funcionará el proyecto. Con ello, se muestran logros y la mayor ventaja de esta práctica es que proporciona gestión y liderazgo de la calidad, a operar y verificar a través del programa (20).
- **Gestión de comunicación:** es el área en donde se definen los procesos y documenta la forma en cómo se hará efectiva la comunicación dentro de los interesados y el equipo de trabajo (20).
- **Gestión de riesgos:** es el área en donde se definen los distintos tipos de riesgos que puedan generarse en el proceso de construcción del proyecto (20).

2.2.7. Scrum

Es una metodología de trabajo iterativa e incremental, sumamente útil y reconocida en estos últimos tiempos, debido a la competencia global y el desarrollo de soluciones bajo un entorno altamente competitivo, es una metodología enfocada al desarrollo de proyectos y productos, en donde una de sus características más notables es el de tratar al cliente o interesado del producto como un miembro más del equipo y no solo un

usuario o cliente, quiere decir que, este interesado lleva a cabo revisiones o retroalimentación después de terminar cada iteración o *Sprint* como se denomina en este marco de trabajo, por lo que es una forma de mantener una alineación más sólida hacia los objetivos del producto y expectativas que tiene el cliente; en resumen, la metodología *Scrum* trata como tema importante las acciones de “inspeccionar y adaptar”, de modo que el producto está en una constante revisión y corrección de los requerimientos originalmente tratados en una primera planificación, llamado también creación de la pila de producto o *Backlog* (21).

Como una información básica y más materializada se mencionan las características principales que esta metodología lleva a cabo.

Como inicio en la metodología *Scrum* el dueño de producto (PO, producto *owner*) define la pila del producto o pila de requerimientos que se realizará a lo largo del proyecto, donde también, en un principio se define la duración de un *Sprint*, donde se denomina *Sprint* a cada iteración del proyecto en donde normalmente se define una duración de dos semanas; una vez terminada la definición de todos los requerimientos continúa la planificación del *sprint*, en donde se toma en cuenta la prioridad de cada requerimiento y se estima el esfuerzo de cada una, dando lugar al primer *Sprint* con las historias de usuario definidas a través de la pila de producto mencionada o también denominada como *Backlog* (21).

Los elementos con los que cuenta *Scrum* son:

- **Roles:** dentro de *Scrum* los roles que cumplen todos los involucrados son:
- ✓ **Equipo:** son el conjunto de profesionales que llevan a cabo el trabajo de solucionar las tareas que se les ha asignado en la planificación del *Sprint*, su responsabilidad principal es la de cumplir con las tareas hasta que el *Sprint* termine; participan activamente en la toma de decisiones y tecnicismos que requiera el proyecto.

- ✓ **Dueño del producto:** es la persona encargada de representar al cliente, la responsabilidad de este actor es la de dar el alcance exacto de los requerimientos y necesidades del proyecto, también junto al *Scrum* máster son quienes toman decisiones frente al proyecto, naturalmente el dueño del producto es y debe ser una persona altamente conocedora del negocio y los objetivos del producto.

- **Scrum máster:** es la persona responsable del equipo y el cumplimiento de las tareas y de las reglas de *Scrum*, se encarga de validar el *backlog*, administra el cumplimiento de los requerimientos y es un enlace entre el cliente y el equipo al momento de tener dependencias de parte del dueño del producto.

- **Artefactos**
 - ✓ **Pila del producto:** es un documento en el que se definen las necesidades y objetivos que se desean en el proyecto, son listadas a través de las historias de usuario, normalmente son elaboradas por el *Scrum* máster y el dueño del producto.
 - ✓ **Pila de Sprint:** es la lista de historias de usuario descompuestas en tareas que se llevan a cabo en la iteración que viene (*Sprint* actual), esta pila la generan el *Scrum* máster junto al equipo de trabajo.
 - ✓ **Incremento:** es el resultado de cada iteración del producto.

- **Eventos**
 - ✓ **Sprint:** es la unidad de iteración que contiene las tareas definidas para ese bloque de trabajo que se desarrolla en un tiempo definido.
 - ✓ **Reunión de planificación de Sprint:** en este evento se determinan las tareas y se puntúan dependiendo qué tan importantes sean y la complejidad que tiene cada tarea, normalmente, este evento sucede cada término del *Sprint* anterior y al inicio del proyecto, definiendo el primer *Sprint*.
 - ✓ **Daily o reunión diaria:** es un evento donde se hacen tres preguntas principales: ¿qué se hizo ayer?, ¿qué se hará hoy? y ¿qué impedimentos tienes?, en donde el *Scrum* máster tiene la información

necesaria para entender el avance diario de cada uno de los integrantes del equipo.

- ✓ **Retrospectiva de Sprint:** es un evento en el que se analizan las tareas terminadas y las tareas que no pudieron completarse, así como también las deficiencias que se tuvo como equipo, para luego dar soluciones o propuestas tácticas para cerrar estas deficiencias en el proyecto.

2.3. Definición de términos básicos

2.3.1. Violencia psicológica

Según la academia española “es el acto extrínseco psíquico sobre un sujeto para imponerle realizar un acto” (22)

2.3.2. Sintagma o frase

Un sintagma o frase significan una o más palabras que están unidas para cumplir una función sintáctica unitaria dentro de la oración (23).

2.3.3. Frases nominales

También conocida como sintagma nominal, es aquel que posee como palabra más relevante un nombre o pronombre, la estructura parte de un sustantivo que se expande mediante diversos modificadores (24), la construcción básica suele ser de la siguiente forma:

$$SN = [\text{Determinantes}] + \text{Sustantivo o sustituto} + \text{adjetivo} \quad (3)$$

Como se demarca en la ecuación, el sustantivo puede tener la siguiente forma, prepuesta con una determinante y pospuesta por un adjetivo (25).

2.3.4. Frases adjetivales

Los sintagmas adjetivales, como lo remarca el nombre, son aquellos que contienen como núcleo un adjetivo y elementos opcionales como modificadores antepuestos o pospuestos entre el núcleo (23). La

estructura simple del sintagma adjetival se podría representar de la siguiente forma:

$$SA = \text{Determinadores} + \text{adjetivo} \quad (4)$$

2.3.5. Frases verbales

El sintagma verbal contiene como núcleo un verbo y también sujetos a los modificadores para que consiga algún contexto en específico, que dependiendo de la clase de verbo, pueden ser distintos modificadores (25).

$$SV = GV + (SN) \quad (5)$$

2.3.6. Frases prepositivas

Se habla de sintagma preposicional cuando se trata de cualquier sintagma u oración precedido de preposición. Es una estructura que siempre se compone de dos funciones (23).

- **Enlace:** que es desempeñada por una preposición o una locución prepositiva.
- **Término:** desempeñada por una palabra, un sintagma o una oración.

$$SP = \text{Enlace} + \text{determinante} + \text{sustantivo} \quad (6)$$

$$SP = \text{Enlace} + \text{adverbio (modificador)} \\ + \text{adverbio(núcleo)}$$

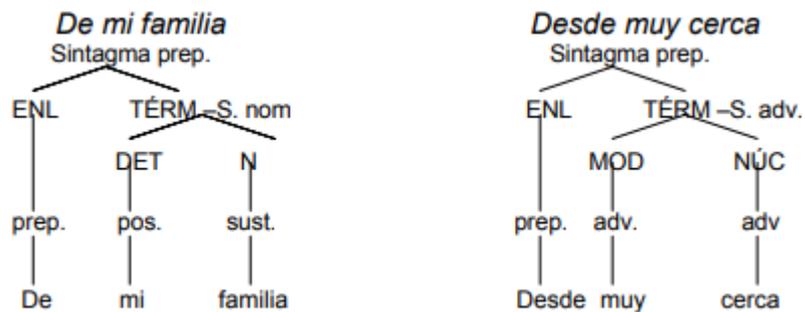


Figura 2. Estructura del sintagma preposicional (23)

2.3.7. Teorema de Naïve Bayes

Es una técnica de clasificación y predicción supervisada útil para determinar estimaciones basadas en el conocimiento subjetivo *a priori*, es decir, que requiere de ejemplos clasificados para que el sistema sea entrenado, sus nombres se deben a las hipótesis tan simplificadas, significa que es independiente condicionalmente de las variables predictoras dada la variable clase (17).

2.3.8. Razonamiento inductivo

Como lo determinó Dávila, “El razonamiento inductivo y deductivo dentro del proceso investigativo en Ciencias Experimentales y Sociales” (26). Determina que esta forma de razonamiento alcanza a una conclusión observando ejemplos y generalizando de ellos a la clase completa, si el sujeto necesita estar absolutamente seguro, necesitará observar todos sus ejemplos.

CAPÍTULO III

METODOLOGÍA

3.1. Metodología aplicada para el desarrollo de la solución

3.1.1. Metodología aplicada a la gestión del proyecto

Para propósitos de orden y buena gestión el proyecto hizo uso de la guía desarrollada por el “*Project Management Institute*” (PMI) el PMBOK, en donde explica el criterio y buenas prácticas para la gestión, para ello y usarlo junto a una metodología ágil, se contemplaron ciertas áreas de conocimiento, como muestra la tabla 6.

Tabla 6. Áreas de conocimiento, gestión del proyecto mediante PMBOK

Área de conocimiento	ítems
Gestión de integración	<ul style="list-style-type: none">• Descripción del proyecto• Objetivos y alcance
Gestión de alcance	<ul style="list-style-type: none">• Desarrollo de pila del producto• Elaboración de EDT
Gestión de tiempo	<ul style="list-style-type: none">• Pila de producto• Definir estructura y duración de <i>Sprint</i>• Estimación de historias de pilas de producto (<i>backlog</i>)
Gestión de costos	<ul style="list-style-type: none">• Planificación de costos
Gestión de calidad	<ul style="list-style-type: none">• Definición de criterios de aceptación de cada historia de usuario de cada <i>Sprint</i>
Gestión de comunicación	<ul style="list-style-type: none">• Definición de <i>dailies</i>• Definición de <i>sprint reviews</i>
Gestión de interesados	<ul style="list-style-type: none">• Identificación de interesados

Como se ve en la tabla 6, el uso de la guía influye directamente en la combinación de la guía con la metodología ágil, considerando puntos y

características de *Scrum* dentro de las áreas de conocimiento de PMBOK, de esta forma se apoyan a las áreas de conocimiento, como el de gestión de comunicación con la candelarización de *dailies*, así como de los *sprint reviews*, en el uso de la definición determinada dentro de cada historia de usuario y, dentro de lo más notable, la gestión de tiempo se incluyó la estimación de historias de usuario y definición de duración de *sprints*.

Para los proyectos de software se busca evidenciar el avance y el cumplimiento de los objetivos que han sido plasmados en etapas anteriores, como el análisis y el diseño del proyecto, este efecto como consecuencia logra mitigar los riesgos, riesgos que pueden surgir en un entorno donde se esperan resultados completos y no escalables, es por lo que se opta el uso de la metodología ágil *Scrum* que brinda un método de trabajo ágil mediante bloques entregables (*sprints*) y revisiones periódicas, en donde se supervisan los avances y complicaciones que han ocurrido durante el *Sprint*, de esta manera el riesgo de generar incertidumbres o riesgos a no poder alcanzar los objetivos se minimiza de forma considerable.

Siguiendo los fundamentos del marco de referencia de metodología ágil, *Scrum* determina ciertos pasos que permiten organizar el proyecto, como se ve en la tabla 7.

Tabla 7. Estructura de tareas según Scrum

Iniciación
<ul style="list-style-type: none"> • Descripción del proyecto • Definición de requerimientos de alto nivel • Definición de arquitectura de software
Planificación
<i>Backlog</i>
<ul style="list-style-type: none"> • Identificación de historias de usuario • Descripción de historias de usuario • Priorización de historias de usuario • Elaboración de <i>sprint backlog</i>
Ejecución
Producto
- <i>Sprint 1</i> (diseño de interfaces y arquitectura del producto)
- Historia de usuario (<i>Painless Project</i>)
PP-01
PP-02
PP-03

PP-04
PP-05
PP-06
PP-07
PP-08

Sprint review 1

- *Sprint 2* (desarrollo)

- Historia de usuario

PP-09
PP-10
PP-11
PP-12
PP-13
PP-14

Sprint review

- *Sprint 3*

- Historia de usuario (desarrollo)

PP-15
PP-16
PP-17
PP-18
PP-19
PP-20
PP-21

Sprint review

- *Sprint 4*

- Historia de usuario (desarrollo)

PP-22
PP-23
PP-24
PP-25
PP-26

Sprint review

- *Sprint 5*

- Historia de usuario (desarrollo)

PP-28
PP-29
PP-30
PP-31
PP-32

Sprint review

3.1.2. Metodología aplicada a la solución de la inteligencia artificial

3.1.2.1. Población

Existe una incalculable cantidad de combinaciones entre palabras para poder determinar exactamente una población de frases agresivas existentes en la realidad peruana, por lo que la población fue determinada infinita, así como también frases que no lo son.

3.1.2.2. Muestra

La muestra obtenida dentro de una población infinita se calculó mediante el uso del muestreo aleatorio simple para poblaciones de un gran número indeterminado, como se puede observar en la fórmula 7, se cuenta con un nivel de confianza al 96% junto a un error de muestreo del 4.1%.

La muestra dio como resultado un total de 623 frases, de las cuales 311 son frases agresivas con insultos y 312 no lo son.

$$n = \frac{z^2 p(1 - p)}{e^2} \quad (7)$$

- n = tamaño de la muestra
- z = coeficiente de confiabilidad 96% (2.05)
- p = 0.50 debido a que no se hizo un estudio poblacional previo
- e = error (0.041)

3.1.2.3. Instrumento de validación de contenido

Para poder hacer uso del inventario fraseológico, elaborado en México por la Universidad de Potosí como referencia (9), fue necesario elaborar un instrumento de validación en donde se emplean las frases con cada tipo de sintagma verbal, adjetival, nominal y prepositiva.

El instrumento de validación se basa en elaborar un conjunto de frases agresivas en donde cada uno de los cuatro entrevistados puntuarán qué tan ofensiva cree que es la frase en un contexto dentro del territorio del estudio. La forma en cómo ha sido validado el contenido fue bajo la propuesta de Hernández Nieto (27), este tipo de validación permite valorar el grado de acuerdo con los expertos entre 3 a 5 con escala de tipo Likert (preguntas con cinco alternativas), la fórmula usada es la siguiente:

$$CVC = \frac{M_x}{Vmáx} \quad (8)$$

La ecuación 8 quiere decir que el coeficiente de validez de contenido es igual a la media obtenida de cada uno de los ítems sobre el valor máximo que se puede obtener en cada ítem, del mismo modo, se debe calcular el error asignado para cada ítem (Pe_i), el cual servirá para reducir el posible sesgo por alguno de los jueces (27).

$$Pe_i = \left(\frac{1}{j}\right)^j \quad (9)$$

El cuestionario elaborado se basa en 36 frases nominales, 19 frases adjetivales, 27 frases verbales y 4 frases prepositivas, donde se entrevistaron a 5 personas coterráneas (los expertos), que responderán en qué grado es ofensiva cada frase dentro del cuestionario que contiene 86 frases en total, dando como resultado un coeficiente de un 81,04%. Como se ve en el anexo 1.

3.1.2.4. Indicadores

El uso de la matriz de confusión ayudó a conseguir indicadores válidos para conocer la exactitud, tasa de error, sensibilidad, especificidad.

A continuación, en la tabla 8 se muestran los indicadores básicos de una matriz de confusión llevados a la práctica en este proyecto de procesamiento de lenguaje natural.

Tabla 8. Indicadores en matriz de confusión

Indicador	Descripción	Instrumento
Verdaderos positivos (VP)	Cantidad de frases detectadas como violentas correctamente	Variable
Verdaderos negativos (VN)	Cantidad de frases detectadas como no violentas correctamente	Variable
Falsos positivos (FP)	Cantidad de frases detectadas como violentas que no lo son	Variable
Falsos negativos (FN)	Cantidad de frases detectadas como no violentas que son violentas	Variable

- Indicador de exactitud

$$E = \frac{VP + VN}{\text{cantidad total de frases}} \quad (10)$$

- Tasa de error

$$TE = \frac{FP + FN}{\text{cantidad total de frases}} \quad (11)$$

- Sensibilidad

$$S = \frac{VP}{\text{total positivos}} \quad (12)$$

- Especificidad

$$ES = \frac{VN}{\text{total de negativos}} \quad (13)$$

3.1.3. Metodología para la recolección de datos

Para el informe, fue oportuno basarse en las estructuras de los sintagmas (frases) registradas en el mencionado trabajo “*Inventario fraseológico de las groserías en los estudiantes de San Luis de Potosí*” (9), señala la forma en cómo están compuestos los insultos y los tipos que existen. Para poder validar las estructuras de este inventario se optó por

adaptarlo a la realidad peruana, empleando sinónimos que, a partir de ello, se hizo una lista de las frases, que a la vez, fue validada por un instrumento de validación, completada por personas residentes en el Perú y que naturalmente conocen las palabras ofensivas o jergas que se utilizan informalmente en sociedad o se entienda como ofensivo. Se consideró elaborar un corpus con sintagmas ofensivos y no ofensivos.

En la tabla 9 se muestran las frases modificadas, siguiendo la estructura morfosintáctica de los insultos registrados en el trabajo antes citado (9).

Tabla 9. Frases consideradas en el instrumento de validación

Frases nominales	
1	Perra cachera
2	Hija de mierda
3	Hija de perra
4	Hija de puta
5	Hija de tu madre
Frases adjetivales	
6	Maldito mocoso
7	Maldita cabeza de mierda
8	Maldita prostituta
9	Maldito perro
Frases verbales	
10	Agárrame esta
11	Cállate el hocico
12	Vete a la concha de tu madre
13	Jálamela
14	Jálatela
15	No estés jodiendo
Frases prepositivas	
16	A la puta madre
17	A la mierda
18	Por la puta madre
19	Por la concha su madre

Como se ve en la tabla 9, se encuentran frases construidas y clasificadas por su tipo de estructura morfosintáctica, las frases de la presente tabla fueron usadas dentro del instrumento de validación.

3.2. Metodología del procesamiento del lenguaje natural

El sistema ha sido desarrollado, basado en el modelo de procesamiento de lenguaje natural propuesto, usando los siguientes niveles del lenguaje natural.

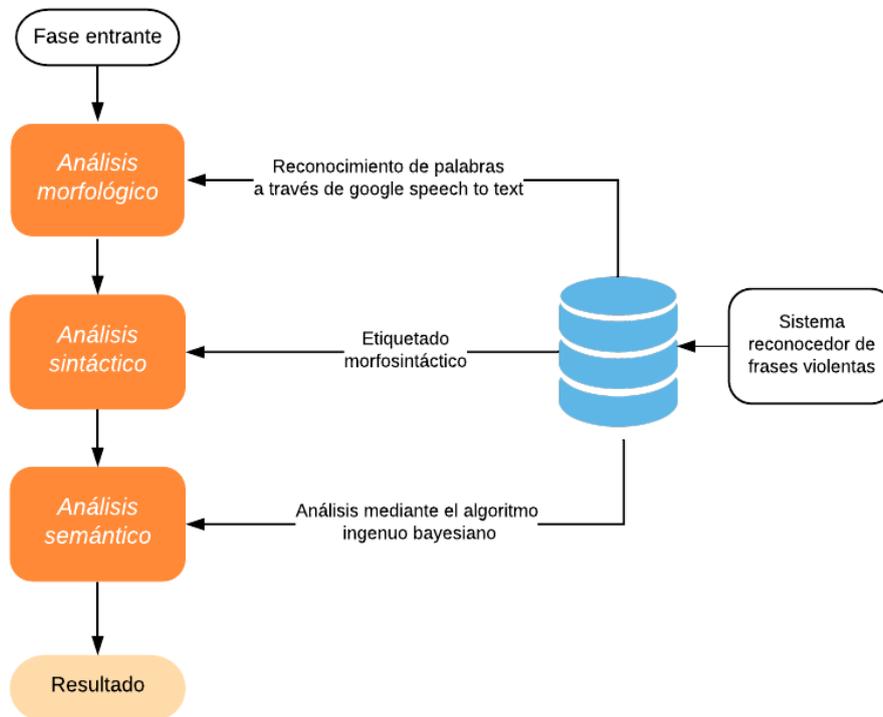


Figura 3. Diseño y flujo del procesamiento del lenguaje natural del sistema

3.2.1. Nivel morfológico

En este proceso, las palabras en conjunto que forman una frase son capturadas a través del programa de *Google Speech to text*, transmitido a través de la API, esta herramienta es capaz de transformar las líneas de audio de voz a palabras escritas en texto, luego de ello, se efectúa la *tokenización* de la palabra: dada la recepción del audio recibido como texto para ser procesado, se eliminan los signos de puntuación y separación de la frase en palabras ("*tokens*"), para la evaluación de cada uno a un nivel léxico.

3.2.2. Nivel sintáctico

A este nivel, el sistema, como lo explica la figura 3, el nivel léxico se encarga de identificar cada palabra con la categoría gramatical que le corresponda, a continuación, se explicará el proceso en la figura 4.



Figura 4. Proceso de sistema de etiquetado

En la figura 4 se muestra la fase final en cómo el sistema entrenado decide generar el etiquetador una categoría gramatical para las palabras que ingresan de una frase.

- 1) **Etiquetador por unigramas:** en este segmento se formó un etiquetador por unigramas basándose mediante un diccionario morfológico “*freeling spanish dictionary*”, que contiene suficiente vocabulario en español con sus etiquetas gramaticales posibles de cada palabra, una deficiencia de esta parte del etiquetador entrenado es el siguiente, ¿qué sucede si la palabra no se encuentra dentro del diccionario?, a continuación, se hará un uso de etiquetador por afijos.

- 2) **Etiquetador por afijos:** se empieza por su definición, un afijo es la secuencia lingüística que se encarga de modificar el significado de un concepto (“palabra”), puede tratarse tanto de un prefijo como un sufijo e incluso un infijo, que se ubica en el centro de la palabra. Lo que permite hacer este etiquetador es el de utilizar todas las palabras del

diccionario morfológico, con el fin de aprender etiquetas más frecuentes según como terminan las palabras.

- 3) **Etiquetador por defecto:** si los etiquetadores anteriores no encuentran coincidencias, ese pondrá uno por defecto, pero esto no quiere decir que sea cualquier etiqueta gramatical, la etiqueta por defecto es la categoría gramatical más recurrente en el idioma español, y es el sustantivo.

3.2.3. Nivel semántico

Este nivel se encarga del análisis del significado de la frase en función a sí misma, es decir, que se encarga de determinar qué tan violenta es la frase ingresada, para esto procesa la frase siguiendo la siguiente lógica en la figura 5.

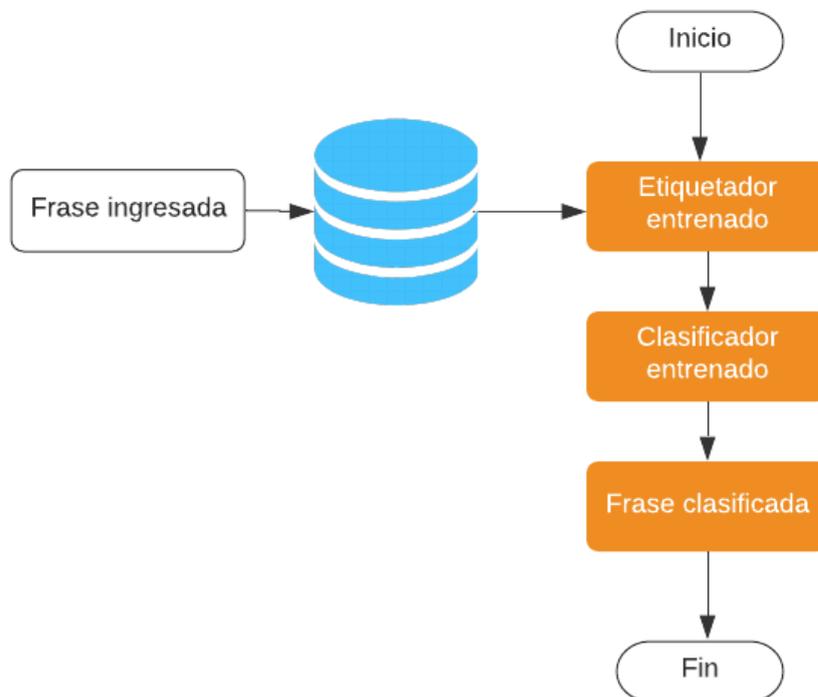


Figura 5. Diseño del análisis pragmático

- 1) **Etiquetador entrenado:** se refiere al etiquetador visto en el nivel léxico, en el cual se forma el etiquetado en ese sistema entrenado previamente, funcional y con una precisión de 97%.

2) **Clasificador entrenado:** el sistema entrenado aplicando el teorema de Bayes, para conseguir una probabilidad independiente de otras variables, es decir, solamente toma en cuenta las palabras y sus respectivas etiquetas.

$$P(a_1, a_2, \dots, a_n | v_j) = P(v_j) \prod_i P(a_i | v_j) \quad (14)$$

Tomada de "*Diseño de un sistema inteligente utilizando procesamiento de lenguaje natural (PLN) para el reconocimiento de mensajes extorsivos*" (13)

Donde:

a_i = palabras del mensaje etiquetadas como verbo, sustantivo, adjetivo o preposición

v_j = categoría, violenta o no violenta

CAPÍTULO IV DISEÑO

4.1. Designación de roles

En la tabla 10 se señala la asignación de roles de los integrantes participantes del proyecto.

Tabla 10. Asignación de roles

N.º de personas	Cargo/rol	Persona encargada	Código
1	Dueño del producto	Martha Luz Cóndor	PO
1	Cliente	Centro de Emergencia Mujer	CEM
1	Equipo de trabajo	Robert Jarod Meza Garcia	RMG
1	<i>Scrum máster</i>	Robert Jarod Meza Garcia	RMG

4.2. Objetivos de la organización

Viéndolo desde un punto de vista administrativo, la implementación de este producto que pretende incluir dentro de sus herramientas de control necesitará un enfoque a la medición y estado de observación frente a agresiones verbales.

El sistema mencionado en este trabajo para poder cumplir con este nivel administrativo debe contar con los siguientes beneficios:

- A. Mejora en la gestión de alertas de violencia verbal y potenciales incidentes de amenazas registradas en el sistema, a través de la aplicación.

B. Alertas con un tiempo más corto al momento de generarse a través de la solución.

C. Mejor seguimiento de abusos verbales dentro de la zona donde se está implementando.

4.3. Definición del alcance

4.3.1. Alcance del producto

- La aplicación permitirá almacenar información de las frases clasificadas y guardar las grabaciones dentro del dispositivo móvil, notificar posibles incidentes mediante un acceso a llamadas de emergencia y dentro de las frases clasificadas al activar el botón.
- El sistema administrador permitirá revisar la actividad de frases clasificadas como violentas mediante un periodo determinado, y puntos en donde se localizan las incidencias.
- El sistema registrará las incidencias y las plasmará en la ubicación que se generó.
- La aplicación móvil podrá detectar frases violentas a través de la voz.
- El sistema del lado del servidor podrá reconocer el evento a nivel de sintagmas y guardar frases más comunes.

4.3.2. Restricciones del producto

- La información solicitada a los *corpus* no ha sido sencilla de recolectar, ya que la seguridad de los datos personales es de prioridad alta en las comisarías, por lo que se controló el experimento con frases creadas previamente, manejando un criterio gramatical para formular frases correctas y frases que pueden confundir al sistema.
- El *corpus* de entrenamiento y pruebas fueron generados a través del instrumento de validación ejecutado, validado con un coeficiente de validez de contenido del 81.04%, siendo considerado que el contenido es apto para el uso del inventario fraseológico de los insultos en la realidad regional en donde el sistema está siendo desarrollado y probado.

4.3.3. Alcance del proyecto

- Los análisis, diseño, desarrollo y pruebas se llevan a cabo dentro de El Tambo, Huancayo, Junín, 2021, probada con gente habitante de la zona.
- Iteraciones de producto
- Documentación de API, aplicativo móvil y sistema
- Informe semanal de avance
- Iteraciones de desarrollo de producto
- Iteraciones de seguimiento y control

4.4. Definición del EDT

En la figura 6 se muestra la estructura de desglose trabajada en el proyecto, mencionando las actividades de desarrollo y revisión al final de cada *Sprint*.

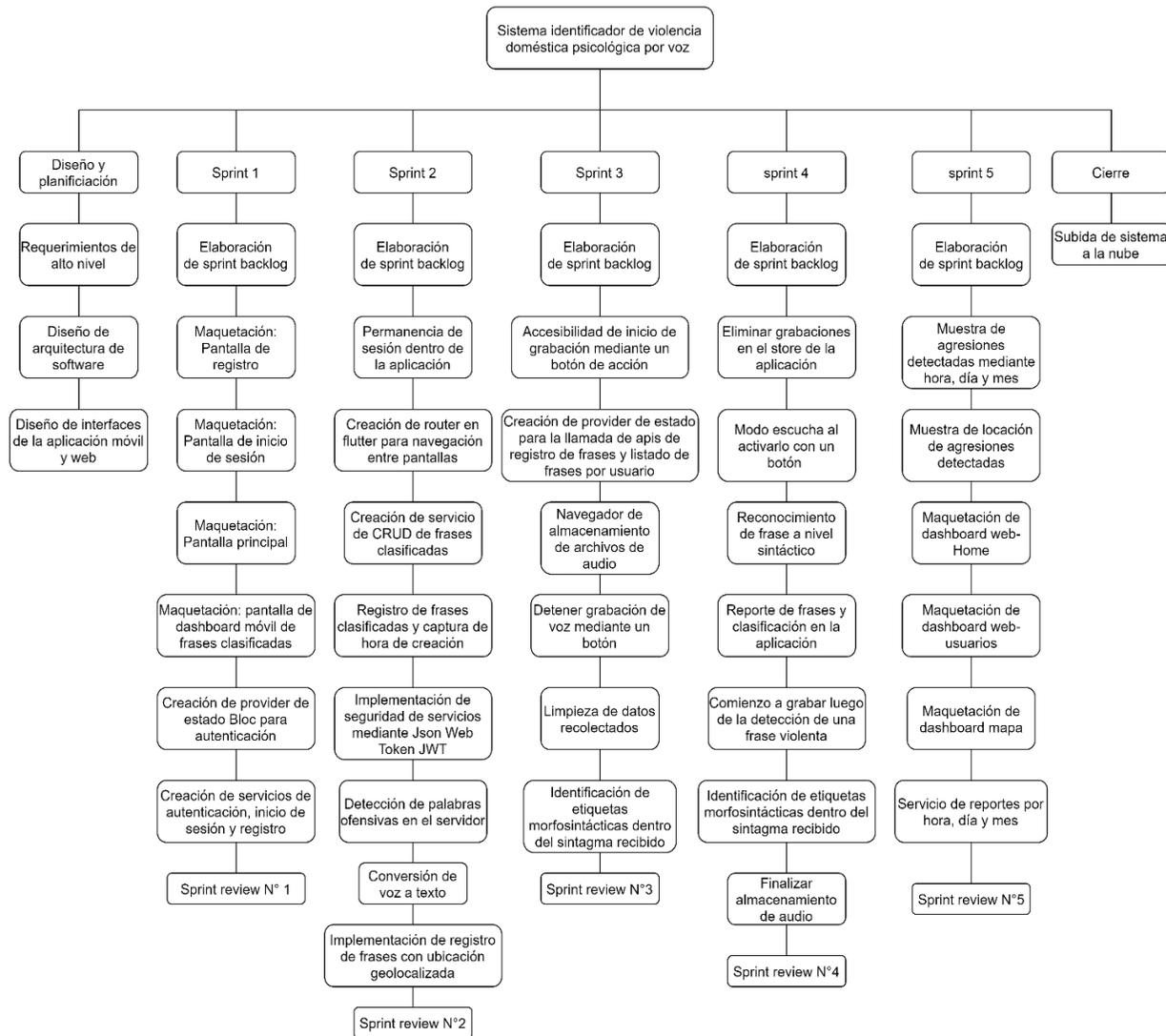


Figura 6. EDT del proyecto

4.5. Identificación de requerimientos

Tabla 11. Pila de requerimientos

Identificador (ID) de la historia	Enunciado de la historia
PP-01	Como un usuario, necesito poder identificar la pantalla de registro, con la finalidad de tener una interfaz en donde registrarme.
PP-02	Como un usuario, necesito poder identificar la pantalla de inicio de sesión, con la finalidad de tener una interfaz en donde pueda iniciar sesión.
PP-03	Como un usuario, necesito poder identificar la pantalla de inicio, con la finalidad de tener familiaridad como en otros aplicativos.
PP-04	Como un usuario, necesito poder observar el historial de frases detectadas, con la finalidad de tener familiaridad como en otros aplicativos
PP-05	Como un usuario, necesito poder observar las funcionalidades de grabadora y activar el modo de escucha, con la finalidad de encontrar un navegador entre pantallas principal.
PP-06	Como un usuario, necesito poder observar las frases que han sido detectadas en la aplicación, con la finalidad de ver el historial de frases.
PP-07	Como un desarrollador, necesito que la información de estado y datos tenga un proveedor de estado global para la autenticación, con la finalidad de tener una aplicación entendible y modificable a través en el tiempo.
PP-08	Como un usuario, necesito que el sistema servidor debe permitir registrar e iniciar sesión, con la finalidad de que los servicios sean seguros y mis datos no se vean vulnerados o editados sin mi consentimiento.
PP-09	Como un usuario necesito poder navegar entre las pantallas de la aplicación, con la finalidad de tener una experiencia de usuario más conveniente.
PP-10	Como un usuario necesito que la sesión de mi cuenta permanezca activa mientras esté la aplicación en mi dispositivo, con la finalidad de no iniciar sesión cada que entre a la aplicación.
PP-11	Como un administrador necesito que la aplicación pueda guardar y visualizar las frases clasificadas, con la finalidad de poder hacer la consulta cuando sea necesario.
PP-12	Como un administrador necesito que la aplicación pueda registrar la hora en que fue clasificada la frase, con la finalidad de poder cuantificar las incidencias por fecha.
PP-13	Como un administrador, necesito que la autenticación del usuario sea posible a través del uso de API, con la finalidad de consultar servicios protegidos a través de ambos clientes (aplicación de celular y aplicación web).
PP-14	Como un usuario, necesito que la aplicación pueda detectar palabras ofensivas, con la finalidad de que se puedan administrar respuestas frente a palabras explícitas.
PP-15	Como un administrador, necesito que la aplicación convierta la voz a texto, con la finalidad de que la frase detectada sea examinada en el lado de servidor.
PP-16	Como un administrador necesito que la aplicación guarde la ubicación de la frase clasificada, con la finalidad de visualizar riesgos o incidentes de violencia psicológica verbal.
PP-17	Como un administrador necesito que la aplicación pueda grabar cuando el usuario lo desee, con la finalidad de que el usuario inicie una grabación en el caso de que necesite grabar lo acontecido.
PP-18	Como un usuario necesito que las grabaciones sean almacenadas en el celular, con la finalidad de tener un registro de mis grabaciones.
PP-19	Como un administrador necesito que la aplicación se comunique hasta los servidores y maneje los datos de manera correcta a través de todas las pantallas.
PP-20	Como un usuario, necesito que las grabaciones pueda verlas dentro de la aplicación, con la finalidad de observar las reproducciones.

PP-21	Como un usuario necesito que la aplicación permita detener la grabación mediante un botón, con la finalidad de terminar la grabación cuando sea necesario.
PP-22	Como un usuario necesito que el modo de escucha pueda activarse mediante un botón, con la finalidad de tener en alerta y escuchar posibles agresiones verbales.
PP-23	Como un usuario necesito que la agresión sea detectada a un nivel semántico, con la finalidad de reconocer las agresiones a un nivel mayor de procesamiento.
PP-24	Como un usuario, necesito que una vez la aplicación detecte una frase agresiva y la aplicación comience a grabar, con la finalidad de grabar la incidencia luego de detectarse una posible agresión.
PP-25	Como un usuario, necesito que las grabaciones puedan eliminarse, con la finalidad de que el usuario tenga el control completo de sus grabaciones.
PP-26	Como un usuario, necesito que las frases clasificadas se guarden en un servidor en internet, con la finalidad de que la data sea permanente y segura.
PP-27	Como un administrador, necesito que las agresiones verbales detectadas puedan verse en un indicador de barras, con la finalidad de visualizar datos en un indicador exacto.
PP-28	Como un administrador, necesito que me muestre las locaciones de agresiones detectadas, con la finalidad de visualizar la información y detectar los periodos en donde se debe dar mayor prioridad frente a emergencias.
PP-29	Como un administrador, necesito que el <i>dashboard</i> esté protegido con un inicio de sesión, con la finalidad de que no todos los usuarios puedan acceder a esa información.
PP-30	Como un administrador, necesito poder agregar usuarios administradores que puedan revisar la información que muestra la página administradora, con la finalidad de poder administrar a los usuarios que puedan acceder en la pantalla administrativa.
PP-31	Como un administrador, necesito poder intercambiar el periodo que me mostrarán las barras (por hora, por mes y año), con la finalidad de visualizar información en cifras.
PP-32	Como un administrador, necesito visualizar las advertencias dentro del mapa con diferente color, con la finalidad de priorizar las advertencias de violencia.

4.6. Pila de producto

Para poder generar las pilas de productos de los *sprints* se tuvo que definir los criterios de aceptación; considerando los requerimientos enlistados se encontró un total de 33, siguiendo la metodología ágil se separaron las pilas en *sprints*, cada una con duración de dos semanas (lo que normalmente se establece).

En la pila de producto se establecen los criterios de aceptación de cada requerimiento establecido con el dueño del producto, en donde especifican los resultados esperados de cada requerimiento.

El detalle de las historias de usuario se establece dentro de las siguientes tablas:

Tabla 12. Pila de producto del Sprint 1

ID de la historia	Enunciado de la historia			Criterios de aceptación	
	Rol	Característica/Funcionalidad	Razón/Resultado	Criterio de aceptación (título)	Resultado/Comportamiento esperado
PP-01	Como un usuario	Necesito poder identificar la pantalla de registro.	Con la finalidad de tener una interfaz en donde registrarme.	Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.
				Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles.
PP-02	Como un usuario	Necesito poder identificar la pantalla de inicio de sesión.	Con la finalidad de tener una interfaz en donde pueda iniciar sesión.	Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.
				Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles.
PP-03	Como un usuario	Necesito poder identificar la pantalla de inicio.	Con la finalidad de tener familiaridad como en otros aplicativos.	Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.
				Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles.
PP-04	Como un usuario	Necesito poder observar el historial de frases detectadas.	Con la finalidad de tener familiaridad como en otros aplicativos.	Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.
				Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles.
PP-05	Como un usuario	Necesito poder observar las funcionalidades de grabadora y activar el modo de escucha.	Con la finalidad de encontrar un navegador entre pantallas principal.	Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.
				Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles.
				Funcionalidad	La vista debe estar disponible para poder acceder a todos los componentes de funciones en la vista.
PP-06	Como un usuario			Diseño y vista intuitiva	La vista debe permitir entender qué campos y cómo se deberán llenar en la aplicación.

PP-07	Como un desarrollador	Necesito poder observar las frases que han sido detectadas en la aplicación.	Con la finalidad de ver el historial de frases.	Campos y textos visibles	La vista debe contener textos y campos suficientemente grandes para ser entendibles
				Informativo	Debe contener información de fecha y hora.
				Funcional	El funcionamiento del proveedor de estado deberá cumplir con la funcionalidad de manejar el estado global de la aplicación.
PP-08	Como un usuario	Necesito que el sistema servidor debe permitir registrar e iniciar sesión.	Con la finalidad de tener una aplicación entendible y modificable a través en el tiempo.	Escalable	El proveedor debe ser desarrollado para contemplar futuros requerimientos y distintos tipos de lógica dependiendo a lo que requiera la aplicación.
				Seguro	La funcionalidad debe evitar vulnerabilidades, como contraseñas visibles y codificación de estas, así también, como rutas protegidas para el uso de servicios exclusivos.
				Funcional	Debe permitir iniciar y registrar sesión que permita el registro y el almacenamiento en la base de datos de la información registrada, así como, permitir iniciar sesión con el usuario registrado.

Tabla 13. Pila de producto del Sprint 2

Enunciado de la historia				Criterios de aceptación	
ID de la historia	Rol	Característica/Funcionalidad	Razón/Resultado	Criterio de aceptación (título)	Resultado/Comportamiento esperado
PP-09	Como un usuario	Necesito poder navegar entre las pantallas de la aplicación.	Con la finalidad de tener una experiencia de usuario más conveniente.	Funcional	Deberá manejar la navegación entre las pantallas de la aplicación.
				Escalable	El enrutador debe ser desarrollado para contemplar futuros requerimientos y distintos tipos de lógica dependiendo a lo que requiera la aplicación.
PP-10	Como un usuario	Necesito que la sesión de mi cuenta permanezca activa mientras esté la aplicación en mi dispositivo.	Con la finalidad de no iniciar sesión cada que entre a la aplicación.	Persistencia de sesión	La aplicación, en caso se cierre por completo la información, deberá contenerse dentro de la aplicación hasta que se indique lo contrario.
				Solicitud de inicio de sesión	Si el usuario no tiene una sesión deberá darle la opción a iniciar sesión.
PP-11	Como un administrador	Necesito que la aplicación pueda guardar y visualizar las frases clasificadas.	Con la finalidad de poder hacer la consulta cuando sea necesario.	Informativo	La aplicación debe mostrar fecha, hora, frase y estado de clasificación.
PP-12	Como un administrador	Necesito que la aplicación pueda registrar la hora en que fue clasificada la frase.	Con la finalidad de poder cuantizar las incidencias por fecha.	Informativo	La aplicación debe de guardar con una precisión de segundos, la frase clasificada.
PP-13	Como un administrador	Necesito que la autenticación del usuario sea posible a través del uso de API.	Con la finalidad de consultar servicios protegidos a través de ambos clientes (aplicación de celular y aplicación web).	Seguro	La aplicación debe estar codificada con un <i>token</i> de validación y encriptada bajo una llave secreta.
				Duración de <i>token</i>	La duración del <i>token</i> debe ser de al menos 2 horas.
PP-14	Como un usuario	Necesito que la aplicación pueda detectar palabras ofensivas.	Con la finalidad de que se puedan administrar respuestas frente a palabras explícitas.	Respuesta rápida	La aplicación deberá no tener una demora más de un segundo al momento de encontrar la palabra.

Tabla 14. Pila de producto Sprint 3

Enunciado de la historia				Criterios de aceptación	
ID de la historia	Rol	Característica/Funcionalidad	Razón/Resultado	Criterio de aceptación (título)	Resultado/Comportamiento esperado
PP-15	Como un administrador	Necesito que la aplicación convierta la voz a texto.	Con la finalidad de que la frase detectada sea examinada en el lado del servidor.	Precisa	La transformación por texto debe contener la mejor probabilidad de reconocimiento de palabras a través de voz.
PP-16	Como un administrador	Necesito que la aplicación guarde la ubicación de la frase clasificada.	Con la finalidad de visualizar riesgos o incidentes de violencia psicológica verbal.	Precisa	La ubicación detectada debe tener una precisión de al menos 10 m a la redonda.
PP-17	Como un administrador	Necesito que la aplicación pueda grabar cuando el usuario lo desee.	Con la finalidad de que el usuario inicie una grabación en el caso de que necesite grabar lo acontecido.	Visible	El botón debe estar en un lugar central y cercano a dar clic.
				Respuesta rápida	Inmediatamente después de presionarlo, debe grabar inmediatamente.
PP-18	Como un usuario	Necesito que las grabaciones sean almacenadas en el celular.	Con la finalidad de tener un registro de mis grabaciones.	Funcional	Las grabaciones deberán almacenarse dentro del almacenamiento del dispositivo.
PP-19	Como un desarrollador	Necesito que la aplicación se comunique hasta los servidores y maneje los datos de manera correcta a través de todas las pantallas.	Con la finalidad de no perder información entre pantallas de la aplicación.	Respuesta rápida	La respuesta deberá manejarse de forma asíncrona.
				Veracidad de información	La data debe permanecer actualizada frente a cualquier cambio desde el servidor.
PP-20	Como un usuario	Necesito que las grabaciones pueda verlas dentro de la aplicación.	Con la finalidad de observar las reproducciones.	Informativo	Las grabaciones deben contener un detalle (hora y fecha, título, duración).
PP-21	Como un usuario			Accesible	El botón de detener debe estar al alcance de la mano y visión del usuario.

Necesito que la aplicación permita detener la grabación mediante un botón.

Con la finalidad de terminar la grabación cuando sea necesario.

Funcional

La grabación deberá detenerse y almacenarse.

Tabla 15. Pila de producto Sprint 4

Enunciado de la historia				Criterios de aceptación	
ID de la historia	Rol	Característica/Funcionalidad	Razón/Resultado	Criterio de aceptación (título)	Resultado/Comportamiento esperado
PP-22	Como un usuario	Necesito que el modo de escucha pueda activarse mediante un botón.	Con la finalidad de tener en alerta y escuchar posibles agresiones verbales.	Accesible	El botón de activar el modo escucha debe estar al alcance de la mano y visión del usuario.
				Funcional	El botón de activar la escucha y debe estar al alcance de la mano y visión del usuario.
PP-23	Como un usuario	Necesito que la agresión sea detectada a un nivel semántico.	Con la finalidad de reconocer las agresiones a un nivel mayor de procesamiento.	Funcional	Las frases enviadas deben ser evaluadas y clasificadas.
				Verificable	El modelo debe botar una precisión de al menos 90% al momento de correr la prueba.
PP-24	Como un usuario	Necesito que una vez la aplicación detecte una frase agresiva, la aplicación comience a grabar.	Con la finalidad de grabar la incidencia luego de detectarse una posible agresión.	Funcional	La respuesta no debe demorar más de 5 segundos.
PP-25	Como un usuario	Necesito que las grabaciones puedan eliminarse.	Con la finalidad de que el usuario tenga el control completo de sus grabaciones.	Funcional	La grabación debería poder borrarse desde la aplicación.
PP-26	Como un usuario	Necesito que las frases clasificadas se guarden en un servidor en internet.	Con la finalidad de que la data sea permanente y segura.	Funcional	Las frases clasificadas deberán ser enviadas a la base de datos de la aplicación.

Tabla 16. Pila de producto Sprint 5

Enunciado de la historia				Criterios de aceptación	
ID de la historia	Rol	Característica/Funcionalidad	Razón/Resultado	Criterio de aceptación (título)	Resultado/Comportamiento esperado
PP-27	Como un administrador	Necesito que las agresiones verbales detectadas puedan verse en un indicador de barras.	Con la finalidad de visualizar datos en un indicador exacto.	Informativo	La aplicación administrativa debe mostrar cantidad de frases agresivas por fecha, hora y ubicación.
PP-28	Como un administrador	Necesito que me muestre las locaciones de agresiones detectadas.	Con la finalidad de visualizar la información y detectar los periodos en donde se debe dar mayor prioridad frente a emergencias.	Informativo	La aplicación mostrará en un mapa los puntos en donde la aplicación ha detectado agresiones verbales.
				visible	El indicador debe ser lo suficientemente grande como para diferenciarse.
PP-29	Como un administrador	Necesito que el <i>dashboard</i> esté protegido con un inicio de sesión.	Con la finalidad de que no todos los usuarios puedan acceder a esa información.	Seguro	No se podrán acceder a las siguientes páginas sin un <i>token</i> de sesión.
				Poco vulnerable	Hacer ingreso a través de URL directas debería devolver a la pantalla de inicio de sesión.
PP-30	Como un administrador	Necesito poder agregar usuarios administradores que puedan revisar la información que muestra la página administradora.	Con la finalidad de poder administrar a los usuarios que puedan acceder en la pantalla administrativa.	Administrable	Se deben poder listar, crear, actualizar y eliminar usuarios.
PP-31	Como un administrador	Necesito poder intercambiar el periodo que me mostrarán las barras (por hora, por mes y año).	Con la finalidad de visualizar información en cifras.	Informativo	La gráfica de barras debe cambiar la información según el periodo que solicita.
PP-32	Como un administrador	Necesito visualizar las advertencias dentro del mapa con diferente color.	Con la finalidad de visualizar las advertencias de violencia.	Informativo	Se debe poder visualizar las advertencias de las frases detectadas como agresivas.

4.7. Planificación de *Sprints*

En la tabla 17 se muestran a las historias de usuario en donde se determina el tiempo estimado que tardará cada una de las tareas, punto de historia y a la iteración (*Sprint*) al que pertenece.

Tabla 17. Planificación de historias de las iteraciones

ID de la historia	Tiempo estimado (días)	Punto historia (<i>poker</i>)	Iteración (<i>Sprint</i>)
PP-01	1	2	1
PP-02	1	2	1
PP-03	2	2	1
PP-04	2	2	1
PP-05	2	3	1
PP-06	2	3	1
PP-07	2	3	1
PP-08	2	3	1
PP-09	1	1	2
PP-10	3	1	2
PP-11	3	1	2
PP-12	2	2	2
PP-13	2	2	2
PP-14	3	1	2
PP-15	2	3	3
PP-16	2	3	3
PP-17	2	3	3
PP-18	2	3	3
PP-19	2	3	3
PP-20	2	3	3
PP-21	2	3	3
PP-22	2	3	4
PP-23	2	3	4
PP-24	6	5	4
PP-25	2	2	4
PP-26	2	2	4
PP-27	2	2	5
PP-28	2	2	5
PP-29	4	5	5
PP-30	2	2	5
PP-31	2	2	5
PP-32	2	2	5

4.8. Análisis de la solución

4.8.1. Propuesta

Se propuso implementar un sistema reconocedor de frases violentas haciendo uso del algoritmo de Naïve Bayes, para la clasificación de cada frase incluida, también se incluyó el uso de servicios *Rests* para la provisión de recursos desde el servidor hacía las aplicaciones tanto móvil y web, éstas fueron desarrolladas con tecnologías modernas para el desarrollo del lado del cliente (*Server side rendering con Next.js y el SDK de flutter*).

Una vez considerados los recursos utilizados, se pasará a detallar el diseño de la arquitectura de la solución y el diseño de interfaces de la aplicación y página administrativa web.

4.8.2. Arquitectura de la solución

Microservicios, la arquitectura a la que está orientada la solución, permite el uso de recursos que el lado servidor (o *backend*) brinda hacia los clientes, mediante el protocolo HTTP, permite utilizar esos recursos para distintas aplicaciones del lado del cliente, como en este proyecto son la aplicación móvil y la aplicación web.

La arquitectura de la solución se puede ver en la figura 7, donde indica cómo los distintos servicios logran comunicarse y almacenar información.

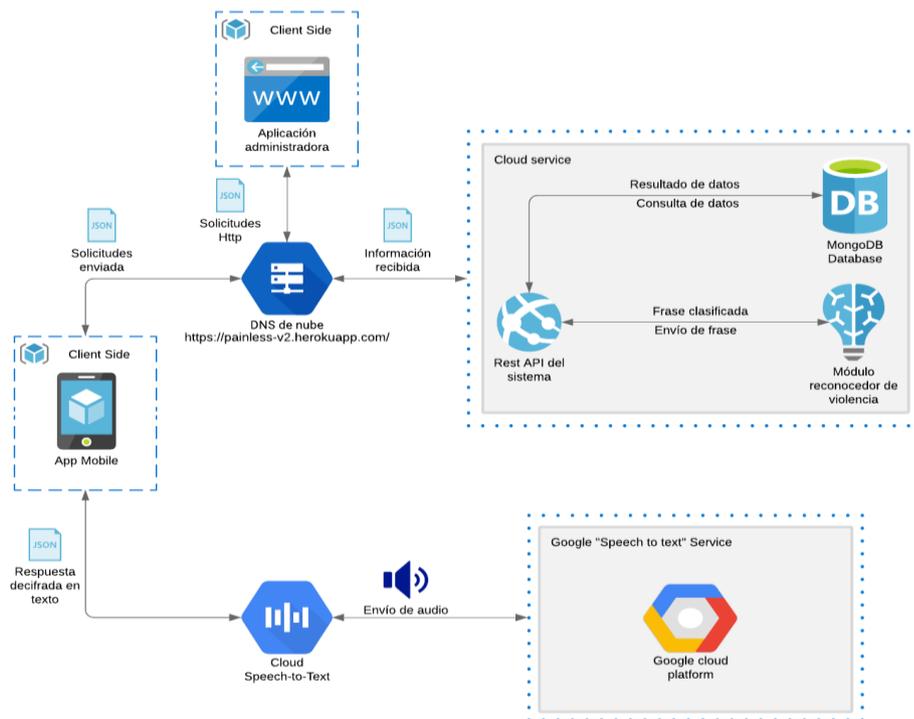


Figura 7. Arquitectura de la solución propuesta

En la figura 7 se ve el diseño de la aplicación, está dividida en tres partes.

4.8.2.1. Lado del cliente

Se creó una aplicación del lado del cliente, la app permitirá a través del micrófono recibir la conversación, una vez registrada será enviada al servicio de "Speech to text" de Google, que permitirá enviar a través de un método GET en la URL asignada la frase convertida en texto, dando como una respuesta de vuelta el resultado de la clasificación de la frase recibida, dependiendo de ello la aplicación activa el micrófono del celular a grabar la conversación.

4.8.2.2. Google cloud, service "Speech to text"

Se hizo uso del servicio capaz de reconocer frases a través de la voz "Speech to text", ya que es un servicio nativo que tienen los Smartphones con sistema operativo Android, de lo que se encargará este lado será recibir el audio capturado en el momento

que el servicio fue solicitado y devolver al lado cliente las palabras detectadas en la grabación.

4.8.2.3. Servicio del módulo reconocedor de frases

Este servicio usa como interfaz la comunicación a través del protocolo HTTP (protocolo de transferencia de hipertexto), que permite realizar una petición de datos y recursos (28). Es así como la aplicación móvil realiza una petición GET, este tipo de petición permite enviar información como consulta dentro de la dirección URL de la aplicación de este, una vez recibida el módulo clasificador determina si la frase fue o no violenta, devolviendo una respuesta y su probabilidad al lado cliente.

4.8.2.4. Servicios *Rest* para consumo de recursos de servidor

Los servicios son la interfaz para el llenado y consulta a la base de datos para el uso de las aplicaciones del lado de cliente, los servicios *Rest* se apoyan de HTTP, haciendo uso de métodos GET, POST, DELETE y PUT principalmente. Los servicios creados con este tipo de interfaz permiten dar una escalabilidad al proyecto según sea necesarios los nuevos requerimientos que sean añadidos.

4.8.2.5. Cliente web

El cliente web consume información desde el servidor consultando la información capturada de la aplicación, permitiendo generar vistas de los indicadores y locaciones de las agresiones detectadas.

4.8.3. Diseño de interfaz gráfica

El proceso de diseño de la interfaz gráfica es una parte importante para un mejor entendimiento de la funcionalidad de la aplicación a los ojos del usuario final, para ello se usaron fundamentos de diseño de interfaces UI y UX, en donde, a través de la data estadística brindada a finales del año 2019 por el MIMP, que facilita indicadores como los números de

casos por tipo de violencia (entre las principales, la psicológica, la física y la sexual), como se observa en la tabla 18.

Tabla 18. Número de violencia registradas en el año 2019

Edad	Tipos de violencia				Total	% según edad
	Económica	Psicológica	Física	Sexual		
0 – 17 años	356	25,214	17,631	12,364	55,565	31
18 – 59 años	470	57,552	51,701	5,523	115,246	63
60 a más años	198	7,469	3,250	157	11,074	6

Como se puede ver en la tabla 18, el mayor número de casos de violencia se reportan entre las edades de 18 años hasta los 59, con un total del 63% del total registrado, por ello, el enfoque va a personas jóvenes y adultas que dispongan de un teléfono móvil, un artículo de Ipsos indica que existen 16 652 000 habitantes peruanos entre los 12 y 70 años que usan un smartphone (29).

En la figura 8 se muestra el diseño de la interfaz del aplicativo, fragmentado en pantallas con un flujo determinado, con títulos descriptivos de cada pantalla. La aplicación consiste en una grabadora básica de la que se tendrá total control como cualquier aplicativo enfocado a la grabación de audios, con el adicional que este tendrá un botón en escucha que permitirá al usuario mantener a la aplicación escuchando mientras esté en un ambiente de posible violencia, en el momento que la aplicación detecte alguna frase agresiva, se activará la acción de grabación por micrófono, posteriormente los archivos guardados podrán ser guardadas en la biblioteca de grabaciones. También contará en un pequeño administrador en el que guardará los textos detectados como violentos y los que no fueron detectados como tales.

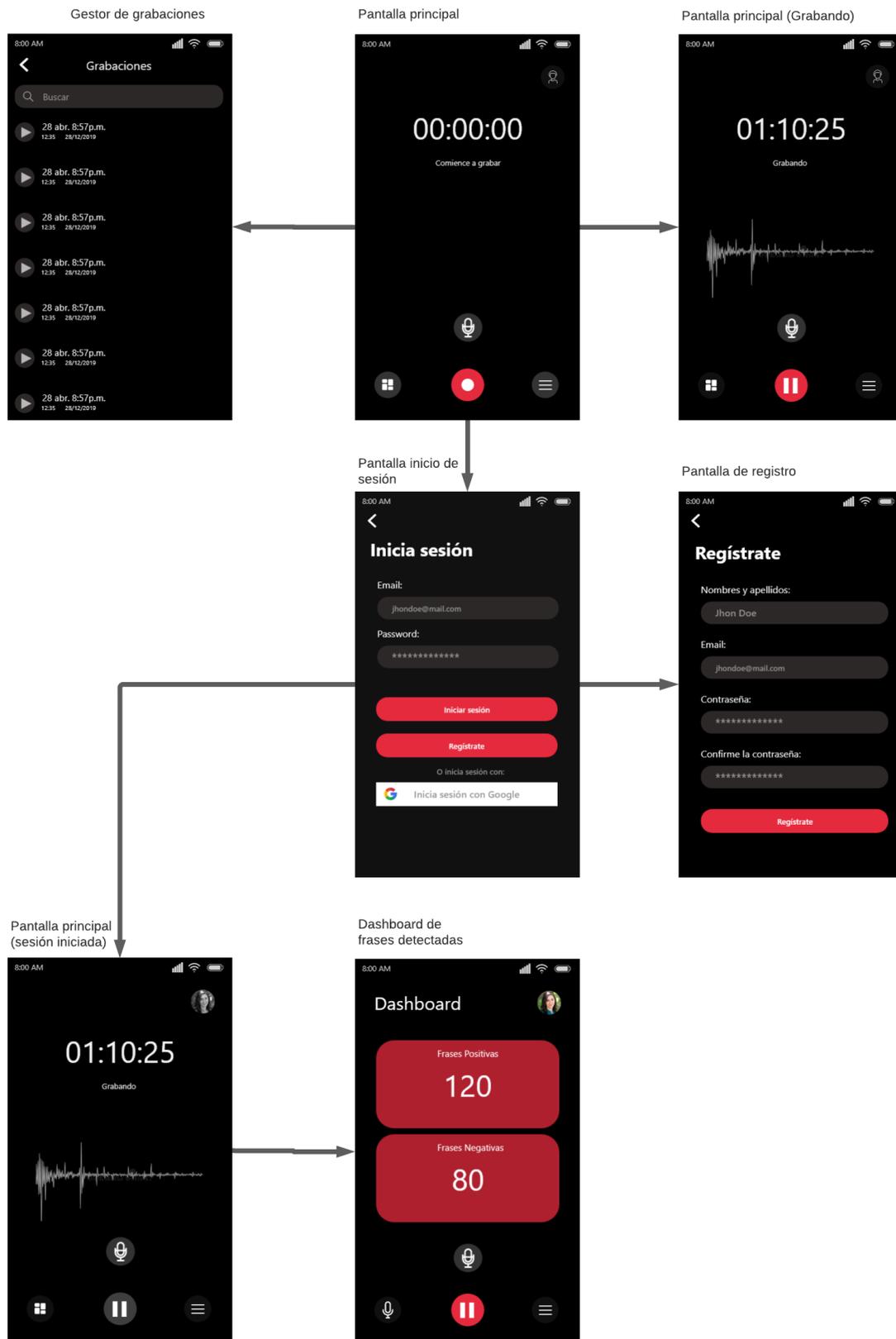


Figura 8. Diagrama de flujo de interfaces de la aplicación

4.8.4. Análisis técnico y económico

Dentro de los costos económicos que se llevaron a cabo en este proyecto se incluirá el costo de despliegue en ambientes de pruebas y ambientes de producción, la subida de la solución de la aplicación en la *Play store de Google* al momento que se acepte la salida a producción y costos indirectos que son contemplados en el proyecto.

Tabla 19. Tabla de costo directo unitario del proyecto

Costo directo unitario				
Concepto	Unidad	Cantidad	Precio unitario	Importe
Alojamiento de instancia EC2	Mes	1	\$ 61.54	\$ 61.54
Certificado SSL	Mes	1	\$ 5.33	\$ 5.33
Costo de instancia por almacenamiento	Año	1	\$ 3.00	\$ 3.00
Costo de dominio "www.painless.agency"	Año	1	\$ 4.99	\$ 4.99
Costo del personal	Mes	1	\$ 750.00	\$ 750.00
Costo directo unitario				\$ 824.86
Costo indirecto				\$ 130.32
Costo unitario				\$ 955.18

Nota: estimación de costo directo, obtenidos a través del registro de gastos del proyecto.

En la tabla 19 se estiman los costos cotizados tanto de los servicios en la nube que se rentarán en *Amazon Web Services*, como el coste de dominios y SSL cotizados en el distribuidor de *GoDaddy* y el salario que se estima del desarrollador del proyecto, dando como un coste directo unitario de \$ 824.86 dólares.

Tabla 20. Tabla de costos indirectos del proyecto

Costos indirectos				
Concepto	Unidad	Cantidad	Precio unitario	Importe
Servicio electricidad	Mes	1	\$35.00	\$35.00
Servicio agua	Mes	1	\$7.00	\$7.00
Servicio de internet	Mes	1	\$40.00	\$40.00
Papelería	Mes	1	\$20.00	\$20.00
Renta	Mes	1	\$500.00	\$500.00
Costos indirectos				\$602.00
Costo directo promedio de producción				\$3,810.45
Porcentaje costo indirecto				16%

Nota: estimación de costo directo, obtenidos a través del registro de gastos del proyecto.

En la tabla 20 se estiman los costos que no influyen directamente en el proyecto, sin embargo, son costos que se consideran debido a que influyen en el desarrollo de manera indirecta, haciendo una estimación mensual total de \$ 602.00 dólares, siendo esto un 16% parte del coste directo promedio de la elaboración del sistema.

Tabla 21. Tabla de costos totales por el proyecto

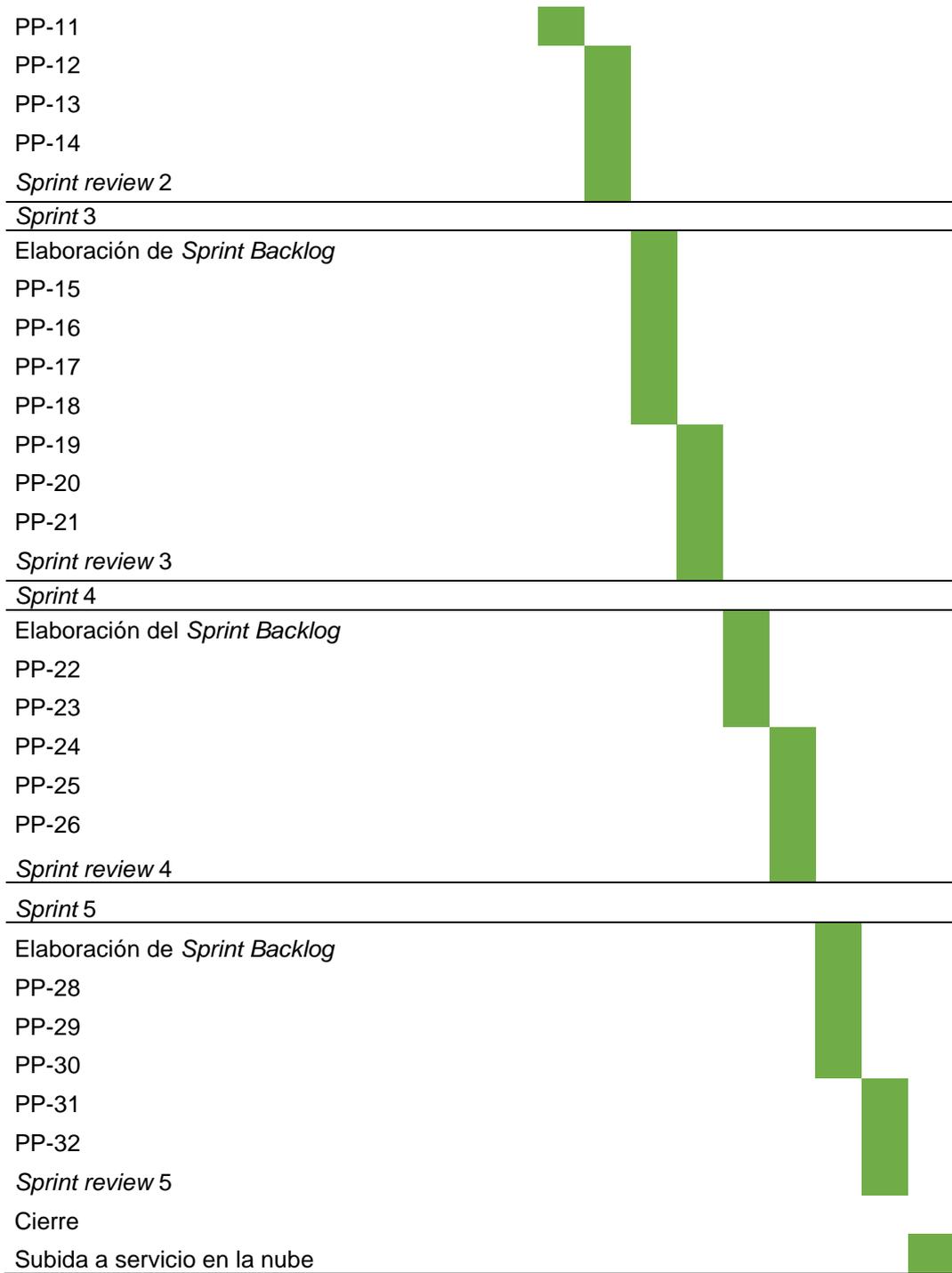
Costo total	
Concepto	Importe
Costo directo	\$3,810.45
Costo indirecto	\$602.00
	\$4,412.45

Nota: estimación de costo directo, obtenidos a través del registro de gastos del proyecto.

En la tabla 21 se evidencia que el costo económico del proyecto ha sido dividido en los costos directos e indirectos con los que contará el sistema, por lo que el costo total fue de \$ 4,412.45 dólares, considerando todos los costos posibles relacionados con el proyecto.

Tabla 22. Cronograma de actividades en base a iteraciones

Entregable	Semana												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Diseño y planificación													
Requerimientos de alto nivel	■												
Definición de arquitectura de software		■											
Diseño de interfaces			■										
Elaboración de <i>product backlog</i>				■									
Sprint 1													
Elaboración de <i>Sprint Backlog</i>				■									
PP-01				■									
PP-02				■									
PP-03				■									
PP-04				■									
PP-05				■									
PP-06				■									
PP-07				■									
PP-08				■									
<i>Sprint review</i>				■									
Sprint 2													
Elaboración de <i>Sprint Backlog</i>					■								
PP-09					■								
PP-10					■								



CAPÍTULO V

CONSTRUCCIÓN

5.1. Construcción

La construcción del sistema de reconocimiento de agresiones consta de dos partes: el lado del servidor (*backend*) y el lado del cliente (*frontend*).

5.1.1. Lado servidor

Para la solución del lado del servidor, se vio el modelo de entrenamiento, se usó el lenguaje de programación *Python* en su versión 3.9, haciendo uso de las librerías: *Flask*, *pymongo* y *NLTK* para el procesamiento de lenguaje natural, así también para el desarrollo de los servicios *Rest* convencionales se utilizó *Typescript* para generar microservicios, con las librerías: *express*, *mongoose*, *morgan*, *@hapi/boom* y más herramientas que facilitan el desarrollo de microservicios confiables y robustos.

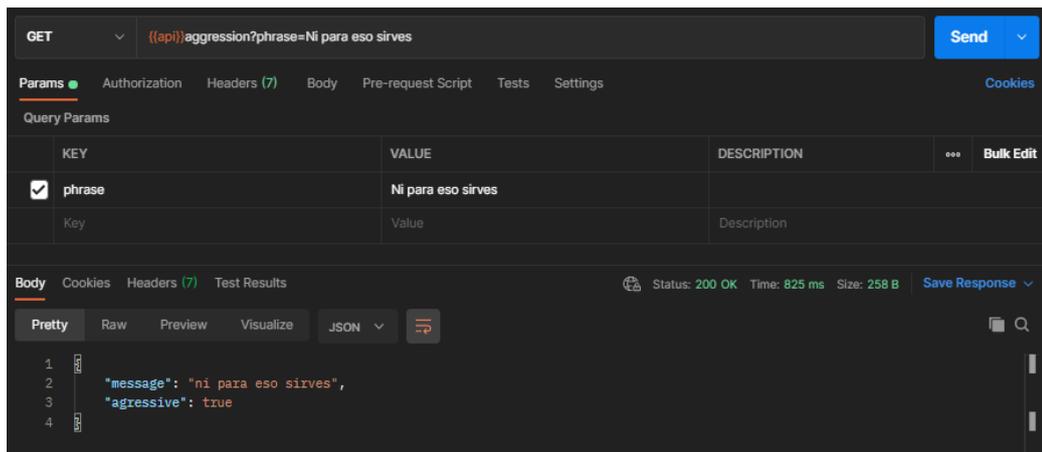


Figura 9. Demostración del envío de información mediante el método GET de http

En el anexo 1 se enseñan fragmentos importantes del código elaborado.

5.1.1.1. Servidor en *Python*

El servidor elaborado en *Python* se encargará todo lo relacionado al procesamiento de lenguaje natural, brindando un servicio dedicado al modelo de entrenamiento por separado.

A. *Flask*

Denominado un micro *framework*, permite poder empezar con proyectos sumamente sencillos y escalar a algo más personalizado, gracias a las herramientas (*plugins*) que se pueden instalar a medida que se necesiten. El servidor creado se enfoca en brindar servicios *restful* sencillos, autenticación, registro de usuarios, almacenamiento de frases y también sirve de interfaz para consultar el modelo entrenado, clasificador de palabras elaborado con el mismo lenguaje.

```

1  from flask import Flask
2  from flask_restful import Api
3  from flask_jwt import JWT
4  from flask_cors import CORS
5  # Own Libraries
6  from src.routes.profile import auth, identity, Profile
7  from src.routes.phrases import ListRoutePhrases, RoutePhrases
8  from src.routes.detect_aggression import Aggression
9  from src.utils.generate_keys import admin_key
10
11 app = Flask(__name__)
12 app.secret_key = admin_key
13 app.config['JWT_AUTH_URL_RULE'] = "/signin"
14 app.config['JWT_AUTH_HEADER_PREFIX'] = "Bearer"
15 CORS(app)
16 api = Api(app)
17 jwt = JWT(app, auth, identity)
18
19 api.add_resource(ListRoutePhrases, '/phrases')
20 api.add_resource(RoutePhrases, '/phrase')
21 api.add_resource(Aggression, '/aggression')
22 api.add_resource(Profile, '/signup')
23
24
25 if __name__ == "__main__":
26     app.run(debug=True)
27

```

Figura 10. Código del servidor web, consumo de servicios creados

B. Pymongo

Es un ORM (*Object Relational Mapping*) que permite la conexión y estructurar una base de datos MongoDB, genera un cliente para realizar consultas a ese tipo de base de datos no relacionales.

```

1  import pymongo
2  ### URL del cliente de la base de datos
3  client = pymongo.MongoClient("*****")
4  db = client.get_database('painless')
5  ### llamado de colecciones de la base de datos
6  db_madwords = db["madwords"]
7  db_users = db["users"]
8  db_phrases = db["phrases"]

```

Figura 11. Conexión con la base de datos MongoDB
 Nota: más sobre el código fuente en los anexos 2, 3, 4 y 5

C. NLTK

NLTK (*Natural language toolkit*) una librería que contiene distintas bibliotecas y *scripts* que ayudan al procesamiento del lenguaje natural, impulsa la investigación y enseñanza con una detallada documentación y *corpus* de software libre.

```
1 import pandas as pd
2 import numpy as np
3 import nltk
4 import random
5 import pickle
6
7 nltk.download("stopwords")
8 from nltk.stem import SnowballStemmer
9 from nltk.corpus import stopwords
10 from nltk.util import ngrams
11
12 nltk.download("punkt")
13 nltk.download("averaged_perceptron_tagger")
14
15 stemmer = SnowballStemmer("spanish")
16 # Importando el etiquetador
17 inputTagger = open("src/pickles/etiquetador-cat-spa.pkl", "rb")
18 tagger = pickle.load(inputTagger)
19 inputTagger.close()
20
21 stop_words = [w for w in set(stopwords.words("spanish"))]
22 punc_lit = [u'.' , u'[' , u']' , u',' , u';' , u':' , u')' , u')' , u'(' , u')' , u'(']
23 stop_words.extend(punc_lit)
24
25 tokenizer = nltk.tokenize.RegexpTokenizer("\w+|[\^\w\|s]+")
26 df = pd.read_csv("src/corpus/corpus.txt",
27                 sep="|",
28                 names=["clase", "contenido"])
29 df["tokens"] = df["contenido"].apply(
30     lambda x: [w.lower() for w in tokenizer.tokenize(x) if w not in punc_l
31     it])
32 df["bigram"] = df["tokens"].apply(lambda x: list(nltk.bigrams(x)))
33 df["fourgram"] = df["tokens"].apply(lambda x: list(nltk.ngrams(x, 4)))
34 df["tags"] = df["tokens"].apply(lambda x: [t for w, t in tagger.tag(x)])
```

Figura 12. Código de entrenamiento, primera parte

5.1.1.2. Servidor en *Typescript*

El servidor elaborado en *Typescript* está enfocado a dar servicios de autenticación, reportes de incidentes, y creación de usuarios, debido a la forma de tratar sus procesos de forma asíncrona, da posibilidad a una mayor capacidad de llamadas o solicitudes al *endpoint* del API, por lo que favorece a la escalabilidad de servicios futuros que requieran un trato mayor en cuanto refiere a velocidad y desempeño, a continuación se explicarán las librerías y el uso que se le dio a estas en el desarrollo.

A. Express JS

Express js es uno de los *frameworks* más usados dentro del desarrollo web actual, provee de una infraestructura totalmente simple que permite agregar más funcionalidades y herramientas a medida que sea necesario dependiendo del sistema que se requiera programar.

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript and shows the setup of an Express.js server. It includes imports for various modules like express, config, passport, and several middlewares. The code is numbered from 1 to 35. The editor has three colored window control buttons (red, yellow, green) in the top left corner.

```
1 import express from 'express';
2 import config from './config';
3 import passport from 'passport';
4 import './database';
5 import {
6   errorHandler,
7   logErrors,
8   wrapErrors
9 } from './middlewares/errorHandler';
10 import notFoundHandler from './middlewares/notFoundHandler';
11 import cors from 'cors';
12 import morgan from 'morgan';
13 import './middlewares/basicMiddlewares';
14 import './middlewares/jwtMiddleware';
15 import routes from './routes';
16
17 const { PORT } = config;
18
19 const app = express();
20
21 app.set('port', PORT);
22
23 app.use(express.json());
24 app.use(cors());
25 app.use(morgan('dev'));
26 app.use(passport.initialize());
27
28 app.use('/api', routes);
29
30 app.use(notFoundHandler);
31
32 app.use(logErrors);
33 app.use(wrapErrors);
34 app.use(errorHandler);
35 export default app;
```

Figura 13. Código de servidor, módulo principal

Como se ve en la figura 13, en el código se puede apreciar la dinámica en cómo se instancian los módulos del servicio dando un orden mayor y un desarrollo enfocado a *middlewares*, esto quiere decir que los procesos intermedios están separados, reunidos y organizados en el *script* principal en un orden

determinado para procesar validaciones de estos *middlewares*, el orden en cómo fue desarrollado está basado en repositorios y servicios, esto quiere decir que las consultas a la base de datos son instanciadas una sola vez y estandarizadas para ser enviadas a ser utilizadas en los módulos de servicios.

B. Mongoose

El ORM por excelencia tanto como para *javascript* y *typescript*, es el *driver* que permite la conexión hacia la base de datos desde la aplicación, permite crear “Schemas”, modelos tipados, similares a las tablas en modelos relacionales, solo que esta vez para las colecciones no relacionales de *mongodb*, dan un orden más seguro dentro de la lógica del negocio.

```
1 import { Schema, Document, model } from 'mongoose';
2 import bcrypt from 'bcrypt';
3 export interface IUser extends Document {
4   email: string,
5   name: string,
6   lastname: string,
7   role: number,
8   password?: string,
9   matchPassword: (password: string) => Promise<boolean>,
10  encryptPassword: (password: string) => Promise<string>
11 }
12
13 const userSchema: Schema<IUser> = new Schema({
14   email: { type: String, required: true },
15   name: { type: String, required: true },
16   lastname: { type: String, required: true },
17   role: { type: Number, required: true, default: 3 },
18   password: { type: String, required: true },
19 }, {
20   timestamps: true
21 });
22
23 userSchema.methods.encryptPassword = (password) => {
24   return bcrypt.hashSync(password, bcrypt.genSaltSync(10));
25 }
26
27 userSchema.methods.matchPassword = function (password) {
28   return bcrypt.compareSync(password, this.password || "");
29 }
30
31 userSchema.methods.toJSON = function () {
32   let user = this;
33   let userObject = user.toObject();
34   delete userObject.password;
35   return userObject;
36 }
37
38 export default model<IUser>('users', userSchema);
```

Figura 14. Modelo del documento de usuario

C. Passport JS

Una librería de autenticación hecha para nodeJS, provee a través de unos módulos denominados estrategias, la facilidad de implementación de una gran posibilidad de tipos de autenticación, brindando modularidad e implementación a través de *middlewares* poco intrusivos. En esta ocasión se dará uso de dos estrategias, una es la autenticación básica para el inicio de sesión y otra es el estándar *Json Web Token* para proteger las rutas privadas del API.

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript and implements a Passport.js strategy for JSON Web Tokens (JWT). It includes imports for Strategy, ExtractJwt, StrategyOptions, passport, config, user, and boom. A StrategyOptions object is defined with jwtFromRequest and secretOrKey. The passport.use method is called with a new Strategy instance, which contains a try-catch block to handle user lookup and authentication logic.

```
1 import { Strategy, ExtractJwt, StrategyOptions } from 'passport-jwt'
2 import passport from 'passport';
3 import config from '../config'
4 import user from '../repositories/UserRepository';
5 import boom from '@hapi/boom';
6
7 const opts: StrategyOptions = {
8   jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
9   secretOrKey: config.SECRET_JWT,
10 }
11
12 passport.use(new Strategy(opts, async (payload, done) => {
13   try {
14     const userFound = await user.findByEmail(payload.email);
15     if (!userFound) {
16       return done(boom.unauthorized(), false);
17     }
18     return done(null, userFound);
19   } catch (error) {
20     done(boom.unauthorized());
21   }
22 })
23 ));
```

Figura 15. Estrategia de autenticación a través de Json Web Token

5.1.2. Lado cliente

5.1.2.1. Aplicación móvil

Para la aplicación móvil se optó por una tecnología capaz de renderizar aplicaciones multiplataforma *Flutter*, un SDK (*software development kit*) que utiliza el lenguaje de programación *dart* para su funcionamiento, permite desarrollar aplicaciones móviles tanto en Android como IOS, sin necesidad de aumentar líneas de código extras o reescritura de funcionalidades, dependiendo el sistema operativo.

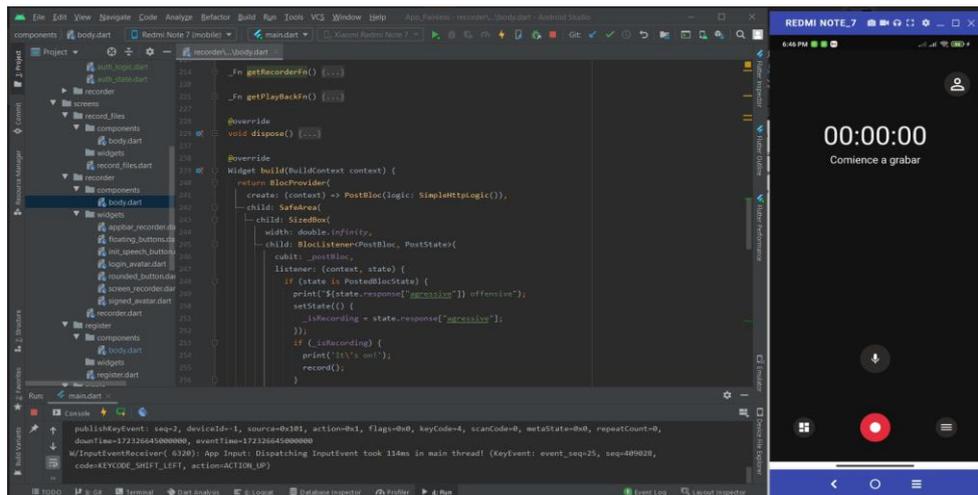


Figura 16. Vista del prototipo móvil en desarrollo

Las librerías utilizadas dentro del desarrollo de la aplicación fueron las siguientes:

A. Flutter sound

Esta librería da todo lo necesario a través de clases y métodos, para el control de los periféricos de audio de un celular, como son el micrófono y las bocinas. Esto permite administrar el momento en el que el micrófono comience a grabar o se detenga, que el reproductor comience la reproducción o detenerlo, así como también observa el nivel de decibeles que el micrófono, la duración de la grabación y la fecha grabada.

B. Speech to text

Librería que permite usar el servicio de *Google Speech to text* (voz a texto), como el nombre del servicio lo indica, el servicio recibe el audio de la frase obtenida, para después devolver en texto las palabras reconocidas.

C. Permission handler

Es un administrador de permisos para apps, verifica si el cliente dio los permisos necesarios para la ejecución de ciertas

funcionalidades, como permisos de uso de micrófono, almacenamiento, *bluetooth*, etc.

D. Flutter secure storage

Es una librería que permite almacenar información de forma segura dentro de la aplicación, se usó para el almacenado de *tokens* de permiso para el servicio *rest* del servidor.

E. Patrón Bloc

Patrón bloc, es una arquitectura de programación que permite un manejo de estado global dentro de la aplicación, es sumamente útil al manejar los *tokens* y estado de resultados que llegan desde el API. Permite mantener un código limpio y organizado para propósitos escalables.

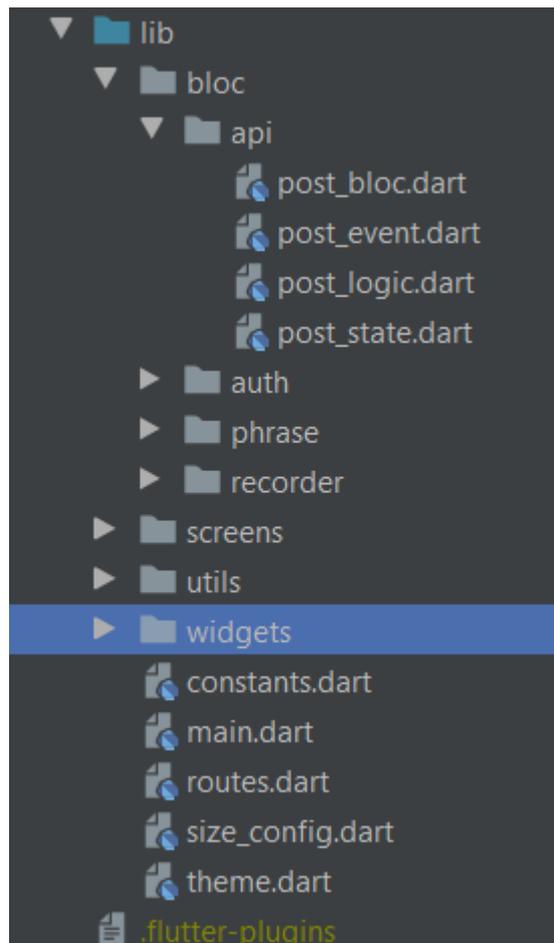


Figura 17. Arquitectura Bloc para la aplicación Painless

En la figura 17 la arquitectura de los ficheros, expresa cómo es que se organiza el trabajo dentro de las carpetas y secciones especializadas en un área en particular, como la carpeta *widgets* almacena componentes de la vista que puedan ser reusables y así no sea necesario reescribir código para un componente similar, en la carpeta de *screens* se pueden observar las pantallas de la aplicación, y la carpeta *bloc* en la que se pueden ver las distintas fuentes de datos que la aplicación obtendrá dependiendo al evento que se ejecute desde el lado de las pantallas a través de los *widgets*.

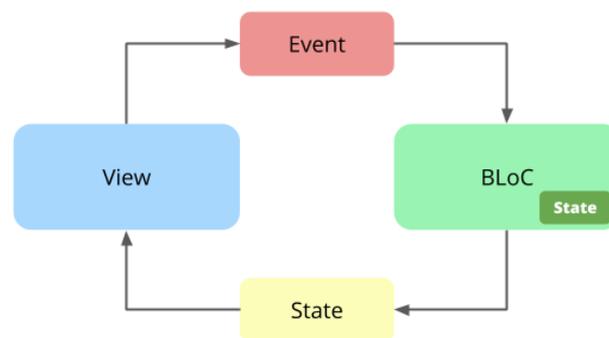


Figura 18. Representación gráfica del patrón Bloc

Como se ve en la figura 18 el lado de la vista envía un evento que es recibido por el *bloc* donde se encuentra la lógica que se necesite, el cual devuelve en un estado a la vista la información como resultado del evento enviado.

5.1.2.2. Aplicación web (lado administrativo)

Para el desarrollo web de la aplicación tuvo en cuenta el uso de programación de una página con un tipo de renderizado desde el lado del servidor (SSR o *Server side rendering*), este tipo de aplicaciones se alojan y compilan desde el lado del servidor, debido a que el servidor es el encargado de renderizar el código en archivos HTML con toda la información solicitada desde el cliente, permitiendo una carga rápida. Con todas las facilidades que brindan las SSR se optó por usar como *framework* principal a la librería NextJs, que permite tener las configuraciones base para

este tipo de despliegue y desarrollo con este tipo de páginas, basado en la librería elaborada por *Facebook React JS*, hace uso del DOM virtual, esto quiere decir que el manejo de eventos se hace de manera más rápida, debido a que los eventos regulares que corren en *Javascript* nativo, se compila desde una capa virtual que no requiere de un refresco a la página o una carga más pesada desde ese nivel.

Para la forma en cómo se organizaron las carpetas, para el desarrollo, se hizo uso de la metodología de diseño “*Atomic Design*”, que permite seccionar de manera específica los componentes de vista en componentes más pequeños (átomos), hasta componentes más significativos (organismos), permitiendo más recursividad y orden al momento de hacer el desarrollo de la aplicación. A continuación, en la figura 19 se podrá ver el orden que guarda cada componente.

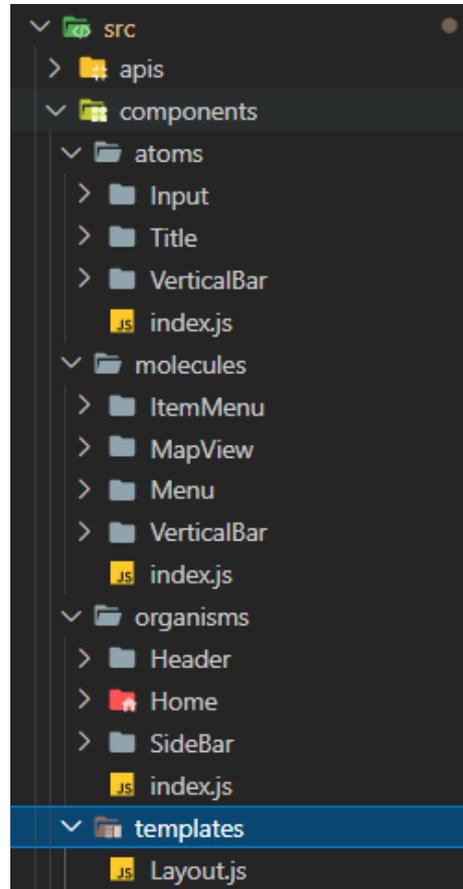


Figura 19. Arquitectura de carpetas para la creación de componentes

Como se ve en la figura 19 se consideró el orden de creación de componentes de forma atómica para asegurar la modularidad y buenas prácticas que se necesitan en un código limpio y entendible, dando como resultado la vista administrativa como se ve en la figura 20.

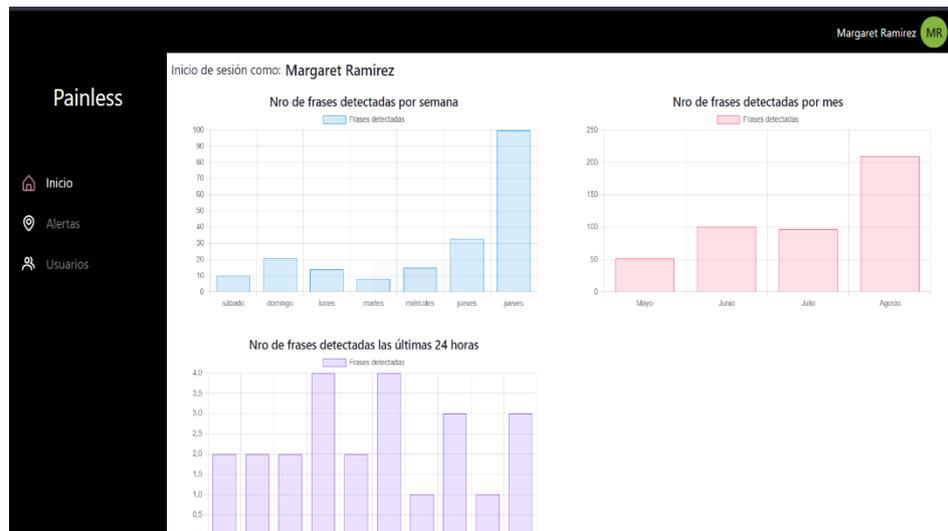


Figura 20. Pantalla administrativa, visualización de cantidad de frases detectadas en el tiempo

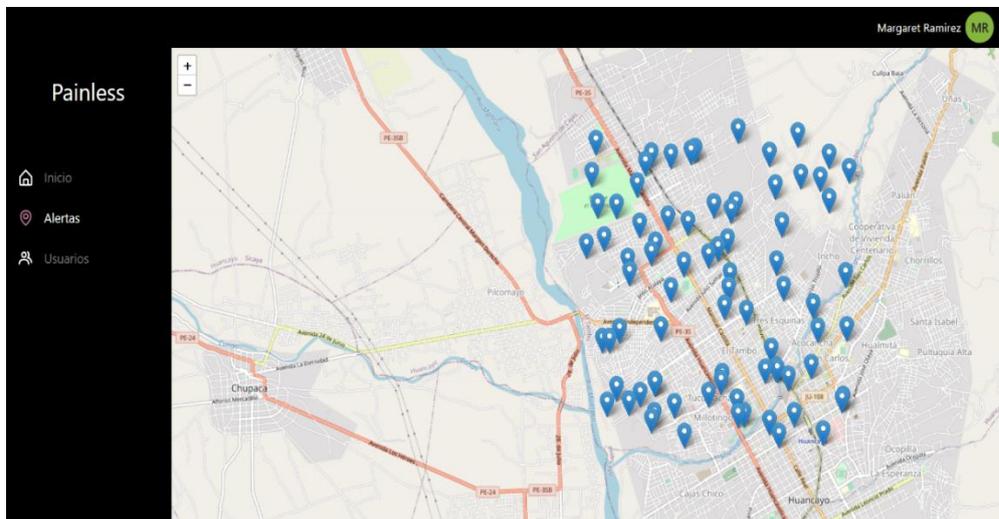


Figura 21. Pantalla de ubicaciones de frases ofensivas clasificadas

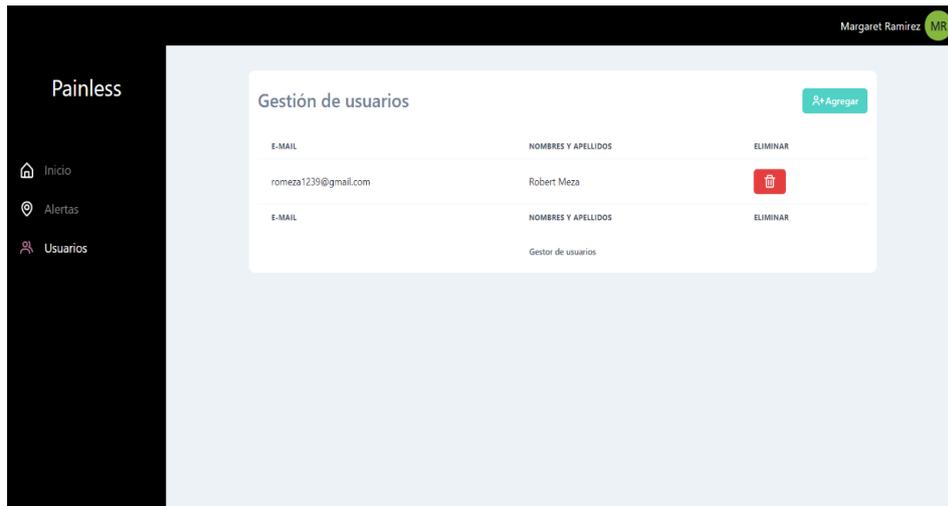


Figura 22. Pantalla de administración de usuarios maestros

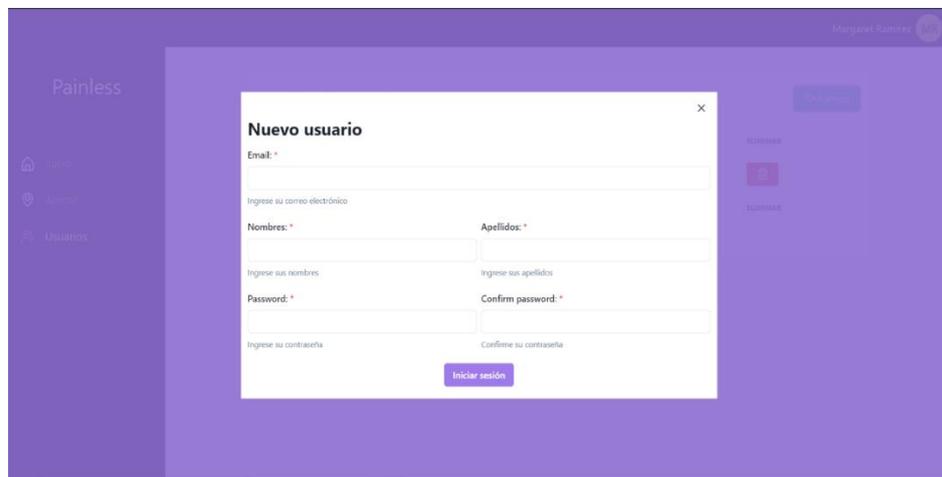


Figura 23. Modal de agregar nuevo usuario

5.2. Pruebas

5.2.1. Pruebas del modelo clasificador

La forma de llevar a cabo las pruebas para determinar la precisión y otras características del modelo entrenado de procesamiento de lenguaje natural fue de la siguiente manera: al sistema se pone bajo un conjunto de pruebas, enfocándose en la recolección de verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos con el propósito de crear una propia matriz de confusión que permitirá medir exactitud, error, sensibilidad y especificidad del sistema.

Tabla 23. Pruebas de frases agresivas

Texto	Resultado
Eres buena para quejarte, hija de puta	Agresivo
Hija de perra, no aguantas sin salir una sola semana	Agresivo
Hija de tu madre, andas de aventurera todo el día	Agresivo
La perra cachera, esa aventurera	Agresivo
Cara de caca, apestas a loco	Agresivo
Sabes que no quiero saber nada, vete a la concha de tu madre	Agresivo
Cojuda de porquería, no necesito más de ti	Agresivo
No me importan tus necesidades, simplemente no me estés jodiendo	Agresivo
Sabes que no vales ni mierda si te vas de aquí	Agresivo
Cierre el hocico y mámamela	No agresivo

Nota: tomada del Programa de pruebas elaborado dentro del sistema

Tabla 24. Tabla de pruebas de frases no agresivas

Texto	Resultado
El amor de la madre no tiene un limite	No agresivo
Eres una desalmada al dejar a tus hijos con esa persona	No agresivo
Desgraciada fue la suerte cuando terminamos	No agresivo
Ni se te ocurra salir de este lugar	No agresivo
Nadie saldrá de aquí, he dicho	No agresivo
Mejor toma un baño de agua fría.	No agresivo
Tu actitud inexpresiva muestra lo falso de tus palabras	No agresivo
Quieres verme, toma el colectivo del comité que está cerca.	No agresivo
Te pido que seas flexible no logro conseguir lo que me pides.	No agresivo
Te portas como cualquier hambriento de la calle.	No agresivo

Nota: tomada del Programa de pruebas elaborado dentro del sistema

Las tablas 23 y 24 representan un fragmento de las pruebas llevadas a cabo, en el anexo 8 y 9, podrá encontrarse la tabla completa de pruebas hechas.

5.2.2. Pruebas del aplicativo

Luego que el modelo entrenado haya sido implementado y utilizable a través del API del servidor, este se une junto a la aplicación móvil, por lo que se hizo una prueba de usabilidad; a continuación se indicará la escala de usabilidad.

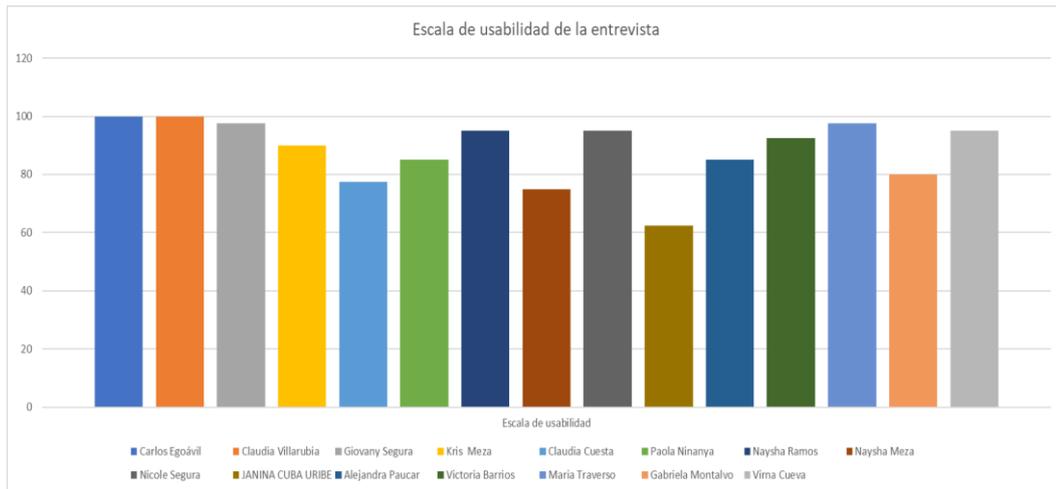


Figura 24. Resultados en escala de usabilidad realizadas en los formularios

Comentan que el uso de la aplicación no fue tan complicado, dando como resultado un promedio de 88.5 en la escala de usabilidad de la aplicación.



Figura 25. Tiempo de respuesta de la aplicación frente a la clasificación

Como se visualiza en la figura 25, se puede observar que la mitad de los usuarios respondieron que existe un retraso mínimo de 1 segundo, teniendo mejores impresiones, el uso de servidores con disponibilidad de producción.

El resultado mostrado luego de un tiempo de uso por los usuarios, mostraron las advertencias en el lado gestor de alertas de la aplicación.

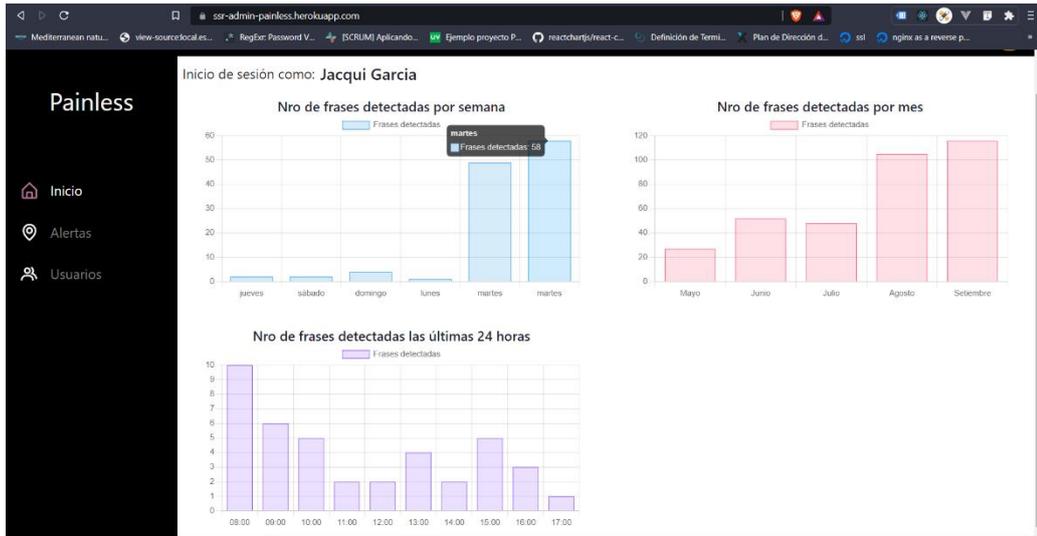


Figura 26. Muestra estadística del uso y detección de frases ubicadas en el tiempo

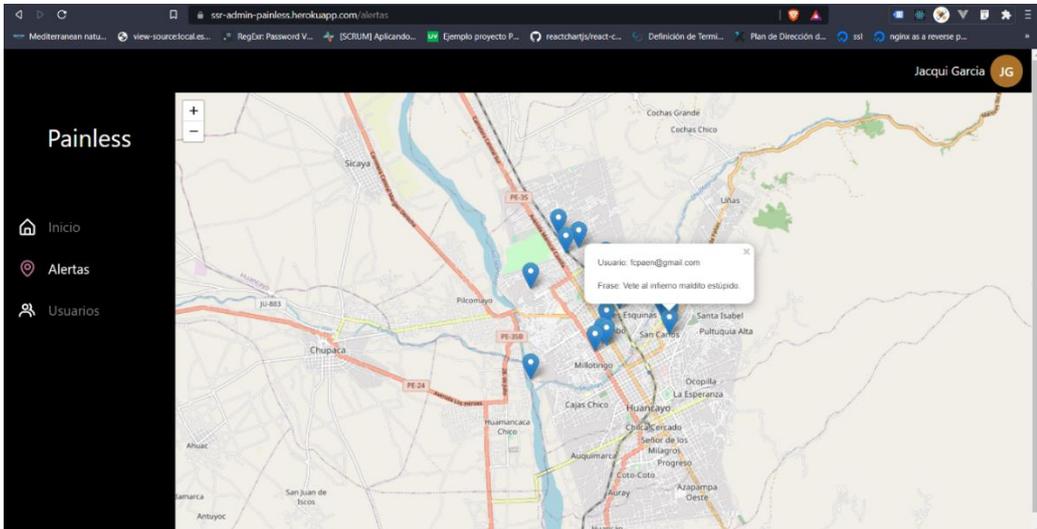


Figura 27. Muestra de las últimas alertas registradas dentro de las 48 horas por cuenta de correo electrónico

5.3. Matriz de confusión, medición de eficacia

Para la medición de eficacia se usará la matriz de confusión en este proyecto, donde se tienen 623 mensajes en total de la muestra, donde 311 son violentas y 312 no lo son, de las cuales un 79.8% [498] de ellos fueron utilizados para el entrenamiento del sistema y el 20.2% [126] para hacer las pruebas.

Tabla 25. Matriz de confusión de resultados de datos de prueba

Matriz de confusión		Clase reconocida	
		Violenta	No violenta
Clase real	Violenta	58	1
	No violenta	5	62

5.3.1. Exactitud de clasificación

$$ACC = \frac{58 + 62}{126}$$

$$ACC = \frac{120}{126}$$

$$ACC = 0.9524$$

Del resultado se interpreta que el modelo de reconocimiento tiene una exactitud (precisión o *accuracy*) del 95,24%.

5.3.2. Tasa de error de clasificación

$$TE = \frac{5 + 1}{126}$$

$$TE = \frac{6}{126}$$

$$TE = 0.0476$$

El modelo de entrenamiento muestra un 4.76% de error.

5.3.3. Sensibilidad de clasificación

$$S = \frac{58}{63}$$

$$S = 0.9206$$

El modelo entrenado tiene un 92,06% de precisión para detectar frases agresivas.

5.3.4. Especificidad de clasificación

$$E = \frac{62}{63}$$

$$E = 0.9841$$

El indicador de especificidad muestra que el modelo entrenado tiene un 98.41% de precisión para encontrar frases no agresivas.

De estos resultados se infiere que el modelo de reconocimiento contiene una precisión aceptable en la que sobrepasa del 90% de exactitud, así también, como de sensibilidad a reconocer frases agresivas, como las que no lo son. Se puede destacar que, de las frases agresivas no detectadas concluyen en frases con adjetivos menos usados, de la misma forma las frases no agresivas.

5.3. Discusión de resultados

Leyendo el trabajo de *Siddhaling Urolagin* (7), usando métodos estadísticos, árboles de decisión, regresión logística y un corpus de información casi infinita, como es la red social Twitter, ha dado resultados hasta de un 99.05% al detectar la fidelización de los usuarios y preferencias hacia otras aerolíneas, probando que el uso de algoritmos y fórmulas matemáticas, el sistema de este proyecto puede conseguir y entender el lenguaje natural de frases (sintagmas) muy cortas, como pueden llegar a ser las agresiones; se evaluaron distintas frases no más extensas de 160 caracteres, haciendo uso de árboles de decisión como lo es el algoritmo de Naïve Bayes y distribución de características frecuentes (palabras, bigramas, bigramas de etiquetas morfosintácticas, etc.); es decir, las que se emplean más en una situación ofensiva y una que no, dando como resultado dentro de un corpus más delimitado, un resultado de exactitud de un 95.24%.

Entendiendo el trabajo de Guzmán en (27), en su análisis de etiquetado automático de su sistema, llega a la conclusión, que al aumentar el diccionario de insultos, el modelo de etiquetado automático del sistema reduce su precisión de detección de instancias ofensivas hasta menos de un 40% y aumenta al momento de detectar instancias no ofensivas hasta un 95%, debido a que los insultos usados del diccionario pueden ser usadas en contextos no ofensivos, de un modo similar el sistema de reconocimiento de agresiones lleva a cabo el uso de detección de los distintos tipos de sintagmas a la que una frase ofensiva puede pertenecer (verbal, adjetival, nominal o prepositiva), al tener una cantidad de frases en el corpus, existe un punto en el que el tipo de sintagmas y palabras empleadas dentro del contexto no resultan ser ofensivas solamente; es por lo que el nivel de exactitud de detección de frases ofensivas se ve reducida

(sensibilidad) a un 92.06% y la especificidad aumenta hasta un 98.41% de especificidad.

De acuerdo con Andrango y Gómez en “*Sistema prototipo actuador por comandos de voz utilizando software libre*” (8), muestran el código y el lenguaje de programación Python utilizado para la detección de comandos; que es un lenguaje de programación libre; actualmente muy usado para el análisis de datos, desarrollo web e inteligencia artificial.

En el trabajo realizado se da a entender que es posible realizar proyectos tan interesantes con software libre y sus librerías con una comunidad grande y activa. En este sistema de reconocimiento de violencia psicológica familiar, no ha sido necesario el uso de tecnologías o software propietario para realizar innovaciones que puedan ayudar en el ámbito social, generando una exactitud de 95.24%.

Argomedeo y Córdor (13), hablan en su trabajo del sistema inteligente detector de extorsiones, y concluyen que es posible una categorización casi perfecta haciendo un debido tratamiento de la información de entrada y el entrenamiento de un número adecuado de corpus de mensajes de texto; al igual como se trata en el trabajo actual, este tuvo el éxito de salir con un porcentaje alto considerando un nivel de confianza del 96%, de donde se extrajo un corpus de un total de 623 con contenido ofensivo y neutral o no ofensivo.

CONCLUSIONES

Habiendo finalizado el proyecto y teniendo en observación la exactitud del detector con un 95.24%, se deja aclarado y se observa lo siguiente:

- Del sistema elaborado en este trabajo, se identificaron los factores de un sistema identificador de agresiones verbales, en donde se concluyó que, el uso de la comunicación de servicios mediante uso del protocolo Http con REST API, que enlazan las aplicaciones con los servidores, también se consideró la usabilidad del sistema y su diseño haciendo uso de los fundamentos de interfaces UX, basándose en las estadísticas reportadas por el MIMP en el 2019, da la información de las personas y su edad con el tipo de violencia que han recibido; de este modo se identificó la disponibilidad de un teléfono móvil según edad, registrada gracias a Ipsos, y se determinó que una implementación oportuna sería a través de un aplicativo móvil, dando un índice de usabilidad de un 88.5%.
- De la investigación bibliográfica se hallan autores como Cortez y otros (14), quienes estudian los niveles de procesamiento de lenguaje natural, además de la construcción del lenguaje originado a través de un conjunto de palabras y afijos; teniendo en cuenta la formación de las palabras, sus variaciones gramaticales y su unión para la creación de una frase. Mientras Durán (9), en su investigación de *"Inventario fraseológico de los insultos"*, se constituye una formalidad de constitución de una frase potencialmente agresiva siendo un insulto, ya que una vez identificada la estructura del etiquetado gramatical se valida e identifica si es o no una frase violenta; en el presente trabajo se comprobó la utilidad de la estructura de una frase ofensiva del trabajo publicado por Durán (9) en la realidad de la región Junín, Huancayo, El Tambo, aplicando un instrumento de validación de contenido, en donde se entrevistaron a cinco personas coterráneas y que el idioma nativo que utilizan es el español, dando como resultado de coeficiente de 81.04%, en donde, según la propuesta de Hernández Nieto (27) el instrumento de validación es aceptable si el resultado tiene como mínimo un 80%.

- Se analizaron frases violentas y frases con la misma estructura fraseológica, reflejando la lógica de los sintagmas del trabajo de Durán (9), por ello se buscó etiquetar gramaticalmente en una primera revisión de las frases y; en una segunda revisión, filtrar palabras con contenido relevante para la evaluación, dando como resultado una clasificación exacta, con una capacidad de detección de frases no agresivas de un 98.41%, siendo mayor a su capacidad de detección de frases agresivas, con un 92.06%.
- Se hizo el diseño del sistema detector de frases violentas, encontrando un algoritmo etiquetador morfosintáctico en español del cual se pudo sacar provecho e implementar a la solución con una precisión en el etiquetado de un 95.24%. Por parte de la conversión del audio a texto se hizo uso de la aplicación inteligente de *Google*, capaz de elaborar tal tarea para poder realizar el análisis solo del texto y, para finalizar, la formulación del algoritmo clasificador fue mediante preentrenamiento con el algoritmo de Naïve Bayes, categorizando con una precisión aceptable de más del 90%.

TRABAJOS FUTUROS

Para la realización de trabajos futuros, se planteó realizar mejoras al algoritmo e incluso realizar una actualización con la nueva solución de palabras hechas vectores, capaces de separarse a un nivel sintáctico, incluso es posible integrar un modelo de soporte vectorial que podría ayudar a resolver el problema de reconocimiento de patrones, del mismo modo, utilizar las frases obtenidas desde la aplicación a través del uso de los usuarios de prueba para realizar la mejora supervisada del sistema clasificador.

La mejora en diseño y funcionalidades de la aplicación móvil para el usuario final dentro del prototipo de aplicación móvil desarrollado estaría dentro de los trabajos pendientes en el futuro.

LISTA DE REFERENCIAS

1. **Ministerio de la Mujer y Poblaciones Vulnerables (MIMP).** *Violencia en cifras.* Lima : s.n., 2019.
2. **Ministerio de la Mujer y Poblaciones Vulnerables.** *Plan de acción para prevenir la violencia contra las mujeres.* Lima : s.n., 2018.
3. **RIVERA, Renzo y CAHUANA CUENTAS, Milagros.** *Influencia de la familia sobre las conductas antisociales en adolescentes de Arequipa-Perú.* 120, 2016, Actualidades en psicología, Vol. 30, págs. 84-96.
4. **RAONA.** Raona. Uso y optimización de micrófono. [En línea] 31 de octubre de 2017. <https://www.raona.com/aplicacion-nativa-web-hibrida/>.
5. **PROYECTOS ÁGILES.** Proyectos ágiles. [En línea] 13 de febrero de 2010.
6. **MESA PÁEZ, Lesley Ofelia, RIVERA LOZANO, Miller y ROMERO DÁVILA, Jesús Andrés.** Descripción general de la Inferencia Bayesiana y sus aplicaciones en los procesos de gestión. *Universidad del Rosario.* [En línea] febrero de 2011. https://www.urosario.edu.co/Administracion/documentos/investigacion/laboratorio/miller_2_2.pdf.
7. **SIDDHALING UROLAGIN, Rida Khan y BITS, Pilani.** *Airline sentiment visualization, consumer loyalty measurement and prediction using twitter data.* 2018, International Journal of Advanced Computer Science and Applications (IJACSA) , págs. 380-388.
8. **ANDRANGO Jaime; GÓMEZ Estevan.** *Sistema prototipo actuador por comandos de voz utilizando software libre.* 2016, Enfoque UTE, págs. 41-54.
9. **DURÁN PÉREZ, Marco Antonio.** *Inventario fraseológico de las groserías en estudiantes de San Luis Potosí.* 2014, Revista de lingüística y lenguas aplicadas, págs. 79-87.
10. **SAGGION, Horacio.** *Procesamiento de lenguaje natural para el análisis de lenguaje subjetivo.* Buenos Aires, Argentina : Redalyc, 2010.
11. **GARCÍA GARCÍA, Úrsula del Milagro.** *Módulo de reconocimiento de voz a texto independiente de locutor para sistemas de diálogo.* Lima : s.n., 2009.
12. **PUELLES GONZÁLEZ, Giancarlo Arturo; QUEZADA ESPINOZA, José Otoniel.** *Automatización del proceso de la toma de pedidos en el restaurante*

'Don Rulo SAC', utilizando una aplicación móvil con reconocimiento de voz soportada por las API de Google. Trujillo : s.n., 2016.

13. **ARGOMEDO PFLÜCKER, Kevin Christian; CÓNDOR RUIZ, Alexander Tito.** *Diseño de un sistema inteligente utilizando procesamiento de lenguaje natural (PLN) para el reconocimiento de mensajes extorsivos.* Trujillo : Biblioteca digital. Dirección de sistemas de informática u Comunicación - UNT, 2015.

14. **CORTEZ VÁSQUEZ, Augusto; VEHA HUERTA, Hugo; PARIONA QUISPE, Jaime.** *Procesamiento de lenguaje natural.* Universidad Nacional Mayor de San Marcos. Lima : Facultad de Ingeniería de Sistemas e Informática, 2009.

15. **MATÍAS RAMOS, Facundo; IGNACIO VÉLEZ, Juan.** *Integración de técnicas de procesamiento de lenguaje natural a través de servicios web.* Universidad Nacional del centro de la provincia de Buenos Aires. Buenos Aires : s.n., 2016.

16. **MALAGÓN LUQUE, Constantino.** Teorema de Bayes. *Universidad Antonio de Nebrija web site.* [En línea] 14 de mayo de 2003. https://www.nebrija.es/~cmalagon/inco/Apuntes/bayesian_learning.pdf.

17. **FERNÁNDEZ, Enrique.** *Análisis de clasificadores bayesianos.* Escuela de Postgrado, Instituto Tecnológico de Buenos Aires. Buenos Aires : s.n., 2004. Trabajo Final de Especialidad en Ingeniería de Sistemas.

18. **ROMÁN, Víctor.** Algoritmos Naive Bayes: Fundamentos e Implementación. [En línea] <https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>.

19. **BARRERO, Giovanni.** ResearchGate: Evaluación de la eficiencia de los modelos machine learning para la predicción de la calidad del software desarrollado en IBM RPG usando la matriz de confusión y las curvas ROC. *Resarch Gate* . [En línea] Enero de 2019. [Citado el: 24 de Junio de 2020.] https://www.researchgate.net/publication/330366207_EVALUACION_DE_LA_EFICIENCIA_DE_LOS_MODELOS_MACHINE_LEARNING_PARA_LA_PREDICION_DE_LA_CALIDAD_DEL_SOFTWARE_DESARROLLADO_EN_IBM_RPG_USANDO_LA_MATRIZ_DE_CONFUSION_Y_LAS_CURVAS_ROC/link/5c3caf73458515a4c725a9.

20. **GUÍA PMBOK 6.** Professional PMP. *Project Manager.* [En línea] 2018. <https://www.todopmp.com/docs/6pmbok.pdf>.

21. **DEEMER, Pete; BENEFIELD, Gabrielle; LARMAN, Craig; VODDE, Bas.** *Información básica de Scrum*. San Francisco : Creative Commons, 2009.
22. **Real Academia Española.** Diccionario del español jurídico. [En línea] [Citado el: 3 de junio de 2020.] <https://dej.rae.es/lema/violencia1>. ISBN/ISSN.
23. **Xunta de Galicia.** Xunta de Galicia Consellería de educación. *Xunta de Galicia Web site*. [En línea] 2014.
<http://www.edu.xunta.gal/centros/iesnumero1ribeira/?q=system/files/Tipos%20de%20sintagmas%201%20Bac.pdf>.
24. **Administración Nacional de Educación Pública.** El sintagma nominal. [En línea] 2014.
<http://www.anep.edu.uy/prolee/phocadownload/materiales/docentes/glosario/El-sintagma-nominal.pdf>.
25. **FERNÁNDEZ LEBORANS, María Jesús.** *Notas sobre el sintagma en la lengua española*. 1983, Dicenda. Cuadernos de Filología Hispánica, pág. 57.
26. **DÁVILA, Gladys.** *El razonamiento inductivo y deductivo dentro del proceso investigativo en Ciencias Experimentales y Sociales*. 2006.
27. **PEDROSA, Ignacio; SUÁREZ, Javier; GARCÍA, Eduardo.** *Evidencias sobre la validez de contenido: avances teóricos y métodos para su estimación*. s.l. : Acción psicológica, 2014.
28. **Mozilla y colaboradores individuales.** Developer Mozilla. [En línea] [Citado el: 26 de noviembre de 2020.]
<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>.
29. **MENDOZA RIOFRÍO, Marcela.** Ipsos Perú. *ipsos.com*. [En línea] 18 de Noviembre de 2019. [Citado el: 9 de Diciembre de 2020.]
https://www.ipsos.com/sites/default/files/ct/publication/documents/2019-11/elcomercio_2019-11-18_04_2.pdf.

ANEXOS

Anexo 1

Tabla de coeficiente de validez de contenido

Ítem	Juez 01	Juez 02	Juez 03	Juez 04	Juez 05	Sx ₁	Mx	CVC _i	P _{ei}	CVC _{tc}
Ítem 1	1	3	3	5	2	14	2.8	0.56	0.00032	0.55968
Ítem 2	5	5	4	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 3	5	5	4	4	4	22	4.4	0.88	0.00032	0.87968
Ítem 4	4	2	3	3	4	16	3.2	0.64	0.00032	0.63968
Ítem 5	5	5	4	3	3	20	4	0.8	0.00032	0.79968
Ítem 6	4	5	5	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 7	5	4	5	3	3	20	4	0.8	0.00032	0.79968
Ítem 8	3	3	4	2	2	14	2.8	0.56	0.00032	0.55968
Ítem 9	5	4	4	4	2	19	3.8	0.76	0.00032	0.75968
Ítem 10	5	5	4	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 11	2	5	4	5	4	20	4	0.8	0.00032	0.79968
Ítem 12	3	5	4	5	3	20	4	0.8	0.00032	0.79968
Ítem 13	3	5	1	3	2	14	2.8	0.56	0.00032	0.55968
Ítem 14	3	4	2	2	1	12	2.4	0.48	0.00032	0.47968
Ítem 15	5	3	3	2	2	15	3	0.6	0.00032	0.59968
Ítem 16	4	4	5	2	1	16	3.2	0.64	0.00032	0.63968
Ítem 17	5	5	5	3	3	21	4.2	0.84	0.00032	0.83968
Ítem 18	5	5	5	2	2	19	3.8	0.76	0.00032	0.75968
Ítem 19	5	5	5	2	2	19	3.8	0.76	0.00032	0.75968
Ítem 20	5	5	4	4	1	19	3.8	0.76	0.00032	0.75968
Ítem 21	5	5	5	4	1	20	4	0.8	0.00032	0.79968
Ítem 22	3	4	5	2	4	18	3.6	0.72	0.00032	0.71968
Ítem 23	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 24	5	5	5	3	2	20	4	0.8	0.00032	0.79968
Ítem 25	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Ítem 26	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968
Ítem 27	5	5	5	2	3	20	4	0.8	0.00032	0.79968
Ítem 28	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 29	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968
Ítem 30	5	5	4	2	2	18	3.6	0.72	0.00032	0.71968
Ítem 31	5	2	5	3	3	18	3.6	0.72	0.00032	0.71968
Ítem 32	5	4	4	2	2	17	3.4	0.68	0.00032	0.67968
Ítem 33	5	3	5	3	4	20	4	0.8	0.00032	0.79968
Ítem 34	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Ítem 35	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 36	5	4	5	4	4	22	4.4	0.88	0.00032	0.87968
Ítem 37	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Ítem 38	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 39	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 40	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 41	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 42	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 43	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 44	5	3	5	3	4	20	4	0.8	0.00032	0.79968
Ítem 45	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 46	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 47	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 48	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 49	5	3	5	2	4	19	3.8	0.76	0.00032	0.75968
Ítem 50	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Ítem 51	5	3	5	2	4	19	3.8	0.76	0.00032	0.75968
Ítem 52	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Ítem 53	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968

Item 54	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Item 55	4	3	5	3	4	19	3.8	0.76	0.00032	0.75968
Item 56	2	3	5	3	4	17	3.4	0.68	0.00032	0.67968
Item 57	3	3	5	5	4	20	4	0.8	0.00032	0.79968
Item 58	3	4	5	4	4	20	4	0.8	0.00032	0.79968
Item 59	5	4	5	4	4	22	4.4	0.88	0.00032	0.87968
Item 60	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Item 61	5	3	5	3	4	20	4	0.8	0.00032	0.79968
Item 62	4	3	5	4	4	20	4	0.8	0.00032	0.79968
Item 63	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Item 64	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Item 65	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Item 66	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Item 67	3	3	4	5	4	19	3.8	0.76	0.00032	0.75968
Item 68	5	3	3	2	4	17	3.4	0.68	0.00032	0.67968
Item 69	4	2	5	2	4	17	3.4	0.68	0.00032	0.67968
Item 70	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
Item 71	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
Item 72	5	3	5	5	4	22	4.4	0.88	0.00032	0.87968
Item 73	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Item 74	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968
Item 75	3	3	5	4	4	19	3.8	0.76	0.00032	0.75968
Item 76	4	3	5	2	4	18	3.6	0.72	0.00032	0.71968
Item 77	5	2	5	3	4	19	3.8	0.76	0.00032	0.75968
Item 78	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Item 79	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
Item 80	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
Item 81	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
Item 82	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968
									CVC _i	0.8104

Anexo 2

Código del sistema web, iniciador del servicio

```
1 from flask import Flask, jsonify, request
2 from flask_restful import Api
3 from flask_jwt import JWT, jwt_required
4 from flask_cors import CORS
5 # Own Libraries
6 from src.routes.profile import auth, identity, Profile
7 from src.routes.phrases import ListRoutePhrases, RoutePhrases
8 from src.routes.detect_aggression import Aggression
9 import src.services.sentiment as st
10 import src.services.tagger as tagger
11 from src.utils.generate_keys import admin_key
12
13 app = Flask(__name__)
14 app.secret_key = "z1Bpi4z20gv0Y0snpwhtJl5fkzajQrDHH1_-EH8UyaahgxujYySIaw"
15 app.config['JWT_AUTH_URL_RULE'] = "/signin"
16 app.config['JWT_AUTH_HEADER_PREFIX'] = "Bearer"
17 CORS(app)
18 api = Api(app)
19 jwt = JWT(app, auth, identity)
20
21 api.add_resource(ListRoutePhrases, '/phrases')
22 api.add_resource(RoutePhrases, '/phrase')
23 api.add_resource(Aggression, '/aggression')
24 api.add_resource(Profile, '/signup')
25
26 @app.route('/')
27 def index():
28     return "<h1>Esta es la aplicación painless</h1>"
29
30 if __name__ == "__main__":
31     app.run()
```

Anexo 3

Código del sistema detector de violencia (tratamiento del corpus)

```
1 import pandas as pd
2 import numpy as np
3 import nltk
4 import random
5 import pickle
6
7 nltk.download("stopwords")
8 from nltk.stem import SnowballStemmer
9 from nltk.corpus import stopwords
10 from nltk.util import ngrams
11
12 nltk.download("punkt")
13 nltk.download("averaged_perceptron_tagger")
14
15 stemmer = SnowballStemmer("spanish")
16 # Importando el etiquetador
17 inputTagger = open("src/pickles/etiquetador-cat-spa.pkl", "rb")
18 tagger = pickle.load(inputTagger)
19 inputTagger.close()
20
21 stop_words = [w for w in set(stopwords.words("spanish"))]
22 punc_lit = [u'.' , u'[' , ']' , u',' , u';' , u'' , u')' , u')' , u' ' , u'(']
23 stop_words.extend(punc_lit)
24
25 tokenizer = nltk.tokenize.RegexpTokenizer("\w+|[\^\w\ ]+")
26 df = pd.read_csv("src/corpus/corpus.txt",
27                 sep="|",
28                 names=["clase", "contenido"])
29 df["tokens"] = df["contenido"].apply(
30     lambda x: [w.lower() for w in tokenizer.tokenize(x) if w not in punc_lit])
31 df["bigram"] = df["tokens"].apply(lambda x: list(nltk.bigrams(x)))
32 df["fourgram"] = df["tokens"].apply(lambda x: list(nltk.ngrams(x, 4)))
33 df["tags"] = df["tokens"].apply(lambda x: [t for w, t in tagger.tag(x)])
34
35 # Doc features pickle
36 doc_features = open('./src/pickles/df.pickle', "wb")
37 pickle.dump(df, doc_features)
38 doc_features.close()
39
40 all_words_corp = [
41     w.lower() for tokenList in df["tokens"] for w in tokenList
42     if w not in punc_lit
43 ]
44
45
46 def arraysToString(joinChar, arrList):
47     return [joinChar.join(item) for itemsList in arrList for item in itemsList]
48
49
50 bigram_tokens = arraysToString("|", df["bigram"])
51
52 fourgram_tokens = arraysToString("|", df["fourgram"])
53
54 freq_bigram_tokens = nltk.FreqDist(bigram_tokens)
55 freq_fourgram_tokens = nltk.FreqDist(fourgram_tokens)
56
57 top_bigram_tokens = freq_bigram_tokens.most_common(200)
58 top_fourgram_tokens = freq_fourgram_tokens.most_common(200)
59
60
61 util_words = [word for word in all_words_corp if word not in stop_words]
62
63 freq_util_words = nltk.FreqDist(util_words)
64 all_words = nltk.FreqDist(all_words_corp)
65
66 all_tags = [t for listTags in df["tags"] for t in listTags]
67 stemmed_words = nltk.FreqDist([stemmer.stem(word) for word in util_words])
68 freq_all_tags = nltk.FreqDist(all_tags)
69
70
71 top_stemmed = stemmed_words.most_common(100)
72
73 top_tags = freq_all_tags.most_common(100)
74 top_words = freq_util_words.most_common(300)
```

Anexo 4

Código del sistema detector de violencia (efectuando el entrenamiento)

```
1
2 def document_features(document):
3     doc_tags = tagger.tag(document)
4     document_tags = [t for w, t in doc_tags]
5     document_words = set(document)
6
7     bigrams = set(nltk.bigrams(document))
8     bigrams_merged = ["|".join(bigr) for bigr in bigrams]
9     fourgrams = set(nltk.ngrams(document, 4))
10    fourgrams_merged = ["|".join(four) for four in fourgrams]
11
12    document_stemmed = set([stemmer.stem(word) for word in document])
13    set_document_tags = set(document_tags)
14
15    features = {}
16
17    for word, i in top_words:
18        features["contains({})".format(word)] = word in document_words
19    for tag, i in top_tags:
20        features["contains_tag({})".format(tag)] = tag in set_document_tags
21    for stemmed, i in top_stemmed:
22        features["contains_stemm({})".format(
23            stemmed)] = stemmed in document_stemmed
24
25    for bigram_word, i in top_bigram_tokens:
26        features["contains_bigram({})".format(bigram_word)] = bigram_word in bigrams_merged
27    for fourgram_word, i in top_fourgram_tokens:
28        features["contains_fourgram({})".format(
29            fourgram_word)] = fourgram_word in fourgrams_merged
30    return features
31
32 fset = [(document_features(texto), clase)
33         for texto, clase in zip(df["tokens"], df["clase"])]
34 random.shuffle(fset)
35
36 train, test = fset[:496], fset[496:]
37
38 classifier = nltk.NaiveBayesClassifier.train(train)
39 classifier.show_most_informative_features(20)
40
41 save_classifier = open("./src/pickles/nb.pickle", "wb")
42 pickle.dump(classifier, save_classifier)
43 save_classifier.close()
44
```

Anexo 5

Código de etiquetador entrenado

```
1 #Almacenamiento de etiquetadores
2 import codecs
3 import nltk
4 import pickle
5 path="./diccionarios/"
6 entrada=codecs.open(path+"fragmento-wikicorpus-tagged-spa.txt", "r", encoding
7 ="utf-8")
8
9 tagged_words=[]
10 tagged_sents=[]
11 tagged_sents_per_unigrams=[]
12
13 cont = 0
14 for linea in entrada:
15     #cont+=1
16     #if cont == 1000:
17     # break
18     linea = linea.rstrip()
19     if linea.startswith("<" or len(linea) == 0:
20         if len(tagged_words)>0:
21             tagged_sents.append(tagged_words)
22             tagged_sents_per_unigrams.append(tagged_words)
23         else:
24             camps = linea.split(" ")
25             forma = camps[0]
26             lema = camps[1]
27             etiqueta = camps[2]
28             tupla = (forma, etiqueta)
29             tagged_words.append(tupla)
30
31 if len(tagged_words)>0:
32     tagged_sents.append(tagged_words)
33     tagged_sents_per_unigrams.append(tagged_words)
34     tagged_words=[]
35
36 diccionario = codecs.open(path+"diccionario-freeling-spa.txt", "r", encoding
37 ="utf-8")
38
39 cont = 0
40 for linea in diccionario:
41     #cont += 1
42     #if cont == 1000:
43     # break
44     linea = linea.rstrip()
45     camps = linea.split(":")
46     if len(camps)>=3:
47         forma = camps[0]
48         lema = camps[1]
49         etiqueta = camps[2]
50         tupla = (forma, etiqueta)
51         tagged_words.append(tupla)
52         tagged_sents_per_unigrams.append(tagged_words)
53
54 default_tagger = nltk.DefaultTagger("NP00000")
55 affix_tagger= nltk.AffixTagger(tagged_sents_per_unigrams, affix_length=-3,
56 min_stem_length=2, backoff=default_tagger)
57
58 unigram_tagger_diccionario = nltk.UnigramTagger(tagged_sents_per_unigrams,
59 backoff=affix_tagger)
60 unigram_tagger = nltk.UnigramTagger(tagged_sents, backoff=
61 unigram_tagger_diccionario)
62 bigram_tagger= nltk.BigramTagger(tagged_sents, backoff=unigram_tagger)
63 trigram_tagger=nltk.TrigramTagger(tagged_sents, backoff=bigram_tagger)
64
65 sortida = open(path+'etiquetador-cat-spa.pkl', 'wb')
66 pickle.dump(trigram_tagger, sortida, -1)
67 sortida.close()
```

Anexo 6

Tabla de resultados, frases no agresivas y diagnóstico del sistema

Clase	Texto	Resultado
-1	El amor de la madre no tiene un limite	No agresivo
-1	Eres una desalmada al dejar a tus hijos con esa persona	No agresivo
-1	Desgraciada fue la suerte cuando terminamos	No agresivo
-1	Ni se te ocurra salir de este lugar	No agresivo
-1	Nadie saldrá de aquí, he dicho	No agresivo
-1	Mejor toma un baño de agua fría	No agresivo
-1	Tu actitud inexpresiva muestra lo falso de tus palabras	No agresivo
-1	Quieres verme, toma el colectivo del comité que está cerca	No agresivo
-1	Te pido que seas flexible, no logro conseguir lo que me pides	No agresivo
-1	Te portas como cualquier hambriento de la calle	No agresivo
-1	Visité la casa y tuve que comprar el pan para los niños	No agresivo
-1	Nos veremos en el juzgado no podemos consensuar	No agresivo
-1	Ha subido la pensión, tú debes medir los gastos	No agresivo
-1	Escribiré todo lo que hiciste en estas hojas que guardaré años	No agresivo
-1	La visita es espectacular, desearía que siempre puedan venir	No agresivo
-1	Tu ego es grande, tu integridad diminuta	No agresivo
-1	No puedes negarme un plato de comida yo te envío el dinero	No agresivo
-1	He comprado los libros de medicina con mi plata	No agresivo
-1	Llegaré puntual a la apelación	No agresivo
-1	Mi siguiente fecha de visita es en septiembre, lo tienes listo para llevarlo	No agresivo
-1	No debes juzgar de esa manera, muestras la fealdad de tus sentimientos	No agresivo
-1	Puede más tu vanidad que guardar el dinero	No agresivo
-1	Te crees el más fuerte por tu tamaño	No agresivo
-1	Yo me esmero para que su educación no se interrumpa	No agresivo
-1	Conseguiste un abogado diligente	No agresivo
-1	No podré enviar los útiles del colegio	No agresivo
-1	Estas insoportable, cuando se te pase el enojo te vuelvo a llamar	No agresivo
-1	Nuestro caso complejo lo resolverán los abogados	No agresivo
-1	Que no vaya en patineta, puede caerse y será tu responsabilidad	No agresivo
-1	Debe estudiar en la universidad aún si es particular	No agresivo
-1	Me pediste que fuera por el camino angosto, se complicó pasar las cosas	No agresivo
-1	Te vio la trabajadora del hogar del vecino	No agresivo
-1	Nuestro hijo es afortunado de tenerte	No agresivo
-1	Pásame tus zapatos ahora	No agresivo
-1	Saca el pincho de la máquina	No agresivo
-1	Cómo eres de fastidiosa, no puedo creer	No agresivo
1	Consíguete una vida, no me fastidies	Agresivo
-1	Lava los platos ahora	No agresivo
-1	Sirve solo para servir	No agresivo
-1	Te pedí una cerveza desobediente, apúrate	No agresivo
-1	Deja de hacerte de rogar, ven conmigo	No agresivo
-1	No necesito más de tus reclamos	No agresivo

-1	Lo sucedido no es normal, es muy de miedo	No agresivo
-1	Lo del canadiense. Por favor, como que vosotros no se tiren allá si pueden	No agresivo
-1	Lando	No agresivo
-1	En serio Randy, creo que este deporte le hizo algo de daño a tu cerebro	No agresivo
-1	Vale	No agresivo
-1	Más quisiera, digo no, no. Estaba haciendo un estudio de mercado para TB-800 alias ciber osito de peluche	No agresivo
-1	¿Tienes una hermana?	No agresivo
-1	¿Cómo se pierde un bebé?	No agresivo
-1	¿Seis mil?	No agresivo
-1	No, no, no, claro que no	No agresivo
-1	Sí Stan, es lo que quiere la gente. No te preocupes, luces genial.	No agresivo
-1	No Kane con K sino	No agresivo
-1	Y luego cuando intenten agradecernos por los clavos, les decimos que no se agradezca porque hicimos lo correcto, y luego les sonreiremos	No agresivo
-1	¡Eso no! Aj si nos pasa algo	No agresivo
-1	Sí, pues claro	No agresivo
-1	Lucen bien muchachos. ¡A patear culos!	No agresivo
-1	Bueno, Conway, ganas este salto	No agresivo
-1	¿Qué? Creía que significaba lo siento. Se ve que por eso aquella noche en el burdel de Carrachi pasó de repente de malísima a mucho peor	No agresivo
-1	¿Y por qué tú y yo no tuvimos algo la última vez que trabajamos juntos?	No agresivo
1	¡Duro con ellos! ¡Amor con compasión!	Agresivo
-1	¿Antes de que me dispares, puedo decirte una cosa?	No agresivo
-1	De mi primo, la que me diste no daba pego, y pensé que con una de verdad me metería más en el papel	No agresivo
62		

Anexo 7

Tabla de resultados, frases agresivas y diagnóstico del sistema

Clase	Texto	Resultado
1	Eres buena para quejarte, hija de puta	Agresivo
1	Hija de perra, no aguantas sin salir una sola semana	Agresivo
1	Hija de tu madre, andas de aventurera todo el día	Agresivo
1	La perra cachera, esa aventurera	Agresivo
1	Cara de caca, apestas a loco	Agresivo
1	Sabes que no quiero saber nada, vete a la concha de tu madre	Agresivo
1	Cojuda de porquería, no necesito más de ti	Agresivo
1	No me importan tus necesidades, simplemente no me estés jodiendo	Agresivo
1	Sabes que no vales ni mierda si te vas de aquí	Agresivo
-1	Cierre el hocico y mámamela	No agresivo
1	Anda jódete allá en tu familia	Agresivo
1	La concha de tu madre no sé qué hacer contigo	Agresivo
1	A la concha de tu puta madre, la próxima te mato	Agresivo
1	No estés jodiendo, que no entiendes que no quiero nada de ti	Agresivo
1	No molestes, vete a joder con tu madre	Agresivo
1	Te juro por la concha su madre que te mato	Agresivo
1	No puedes venir, vete a la concha de tu madre	Agresivo
1	Estúpido animal rastrero, das vergüenza	Agresivo
1	Cuando no, tu hija de ramera	Agresivo
1	Cojuda buena para nada	Agresivo
1	No mereces vivir, eres una arreacha mala madre, vas a pagar con tu vida	Agresivo
1	Lava mi ropa cara de puta, sirve para algo	Agresivo
1	No sirves ni para limpiar, eres un parásito, te mandarían a mejor vida	Agresivo
1	Eres una pendeja, cara de pinga, en eso no más piensas	Agresivo
1	Como jodes, no puedes dejar de cerrar la boca	Agresivo
1	No aguanto tu inconsciencia de mierda	Agresivo
1	Toda tu vida has sido un mantenido, lárgate o te corto el cuello	Agresivo
1	Puras huevadas hablas, vete a la mierda	Agresivo
1	Apestas cara de caca	Agresivo
1	Pásame mis zapatos cara de chucha, ni para eso sirves	Agresivo
1	Tráeme esas cosas chucha tu madre	Agresivo
1	No me jodas, mis huevos te voy a clavar carajo	Agresivo
1	Eres una mierda siempre me excretas tus pendejadas	Agresivo
1	No me enviaste, puta madre, me jodiste	Agresivo
1	Te voy a matar, mejor vete a pasear con tu amante	Agresivo
1	No vienes, todo se va a la mierda y no ves más a mis hijos	Agresivo
1	Eres más pesada, maldita gorda deja de tragar	Agresivo
1	No te escondas, concha tu perra madre, que te encuentre te mato	Agresivo
1	Te gusta salir mucho cabeza de verga, te voy a asesinar	Agresivo
1	Me cagaste hija de puta, la pagarás con tu vida	Agresivo
-1	Cholo de porquería, cánsate de mi	No agresivo
1	Maldito cholo, muérete	Agresivo

1	Vete al infierno maldito estúpido	Agresivo
1	Por Dios, perro maldito, ya te oigo	Agresivo
1	Intento decir que deberíamos tener diez ligas de sarcasmobol cabeza de pinga y así todos pueden jugar. ¡Ah, mierda!	Agresivo
1	Que, es todo para esa hija de mierda	Agresivo
1	Así mejor, que hay un pedazo de mierda	Agresivo
1	Idiotas, hijos de puta, conmigo está el presidente de Brasil y su esposa. Y miren, aquí, sucia, sucia. Toda, todas las armas al suelo	Agresivo
1	¿Venezolano negro? Pedazo de mierda	Agresivo
-1	Bueno, idiotas. Gracias por ayudarme a pillar el disco. Los rusos me van a dar mucho por él	No agresivo
1	Eh idiota, que soy chofer	Agresivo
1	¿Qué maldito mocoso?	Agresivo
1	No jodas maricón violador de cabras	Agresivo
1	Sucios hijos de puta traicioneros. ¿Quieren dejarme fuera? Me robaron la fórmula y luego intentaron joderme	Agresivo
1	La puta que te parió	Agresivo
1	¿Cómo es que sigues teniendo dos manos? ¡Hijo de puta!	Agresivo
1	Eres idiota por no saber matemática	Agresivo
-1	Este juego es estúpido. No tengo ni idea de cómo salir al campo y ser amable con los del otro equipo	No agresivo
1	No me jodas, Mani	Agresivo
1	¿A quién mierda le importa? No, lo pregunto de verdad	Agresivo
1	Alguien está en la cabeza de todo esto y alguien tiene que pagar. Solo estoy pidiendo una maldita retribución para mi maldito abuelo	Agresivo
1	Porque menudo imbécil, ¿verdad?	Agresivo
-1	Lo invito a lamernos las pelotas, señor	No agresivo
58		

Anexo 8

Prueba de funcionalidad del envío http mediante el método GET hacia el servidor

The screenshot displays the Android Studio interface for a Flutter application named 'App_Painless'. The main editor shows the `body.dart` file with the following code:

```
48 bool signed = false;
49 StreamSubscription _reorderSubscription;
50 double minSoundLevel = 50000;
51 double maxSoundLevel = -50000;
52 String lastWords = '';
53 String lastError = '';
54 String lastStatus = '';
55 String _currentLocaleId = '';
56 int resultListened = 0;
57 bool _isRecording = false;
58 PostBloc _postBloc = PostBloc(logic: SimpleHttpLogic());
59 AuthBloc _authBloc = AuthBloc(authLogic: JWTAuth());
60 PhraseBloc _phraseBloc = PhraseBloc(phraseLogic: HttpPhraseLogic());
61 bool created = false;
62 List<LocaleName> _localeNames = [];
63 String _recorderTit = "00:00:00";
```

The console output shows the following logs:

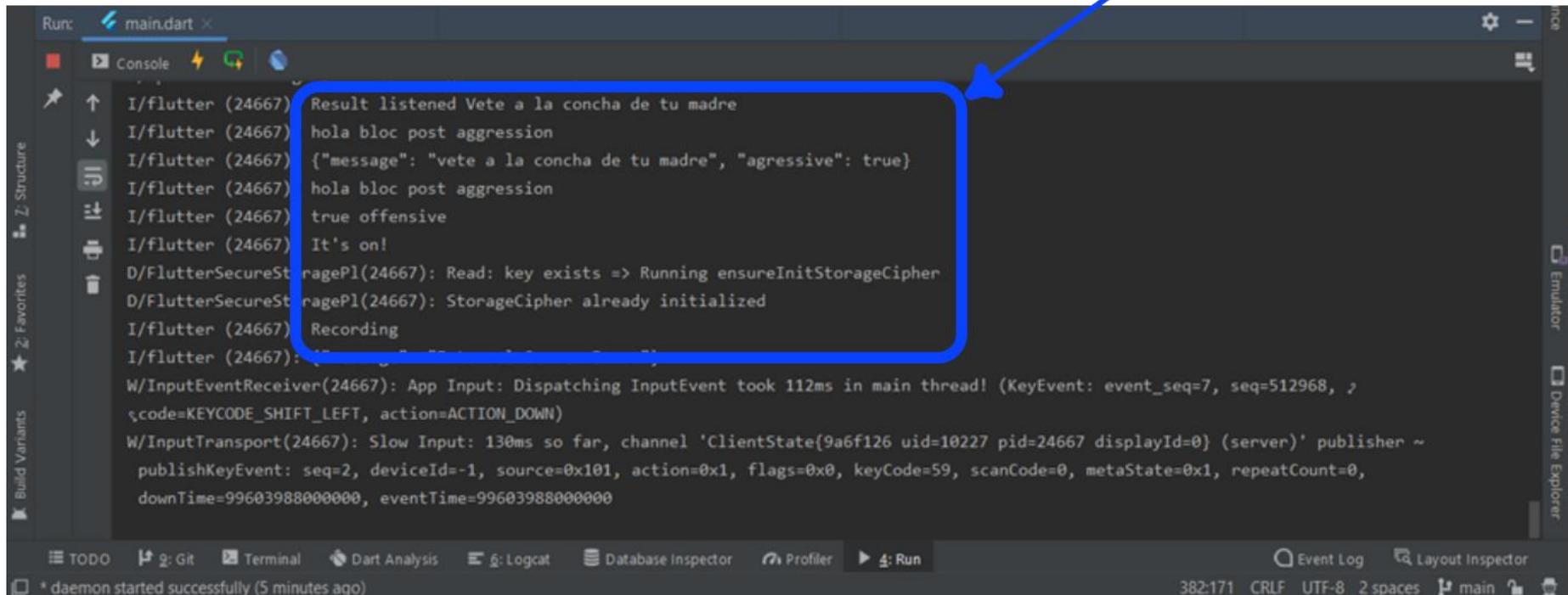
```
I/flutter (24667): Result listened Vete a la concha de tu madre
I/flutter (24667): hola bloc post aggression
I/flutter (24667): {"message": "vete a la concha de tu madre", "agressive": true}
I/flutter (24667): hola bloc post aggression
I/flutter (24667): true offensive
I/flutter (24667): It's on!
D/FlutterSecureStoragePl(24667): Read: key exists => Running ensureInitStorageCipher
D/FlutterSecureStoragePl(24667): StorageCipher already initialized
I/flutter (24667): Recording
I/flutter (24667): {"message": "Internal Server Error"}
W/InputEventReceiver(24667): App Input: Dispatching InputEvent took 112ms in main thread! (KeyEvent: event_seq=7, seq=512968,
vcode=KEYCODE_SHIFT_LEFT, action=ACTION_DOWN)
W/InputTransport(24667): Slow Input: 130ms so far, channel 'ClientState{9a6f126 uid=10227 pid=24667 displayId=0} (server)' publisher ~
publishKeyEvent: seq=2, deviceId=-1, source=0x101, action=0x1, flags=0x0, keyCode=59, scanCode=0, metaState=0x1, repeatCount=0,
downTime=99603988000000, eventTime=99603988000000
```

The virtual device (REDMI NOTE_7) shows a recording interface with a large digital clock displaying '00:24:15' and the text 'Grabando' (Recording). The interface includes a microphone icon and a red recording button.

Anexo 9

Información del log resaltada de petición de servicio de clasificación de violencia

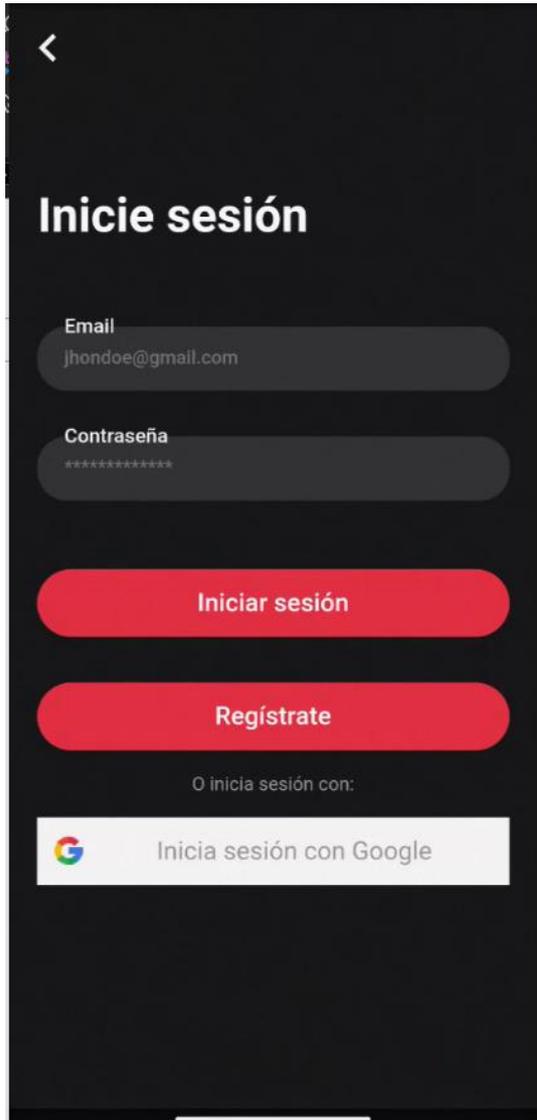
Información log del envío y respuesta del API



```
Runc: main.dart x
Console
I/flutter (24667) Result listened Vete a la concha de tu madre
I/flutter (24667) hola bloc post aggression
I/flutter (24667) {"message": "vete a la concha de tu madre", "agresive": true}
I/flutter (24667) hola bloc post aggression
I/flutter (24667) true offensive
I/flutter (24667) It's on!
D/FlutterSecureStoragePl(24667): Read: key exists => Running ensureInitStorageCipher
D/FlutterSecureStoragePl(24667): StorageCipher already initialized
I/flutter (24667) Recording
I/flutter (24667):
W/InputEventReceiver(24667): App Input: Dispatching InputEvent took 112ms in main thread! (KeyEvent: event_seq=7, seq=512968,
code=KEYCODE_SHIFT_LEFT, action=ACTION_DOWN)
W/InputTransport(24667): Slow Input: 130ms so far, channel 'ClientState(9a6f126 uid=10227 pid=24667 displayId=0) (server)' publisher ~
publishKeyEvent: seq=2, deviceId=-1, source=0x101, action=0x1, flags=0x0, keyCode=59, scanCode=0, metaState=0x1, repeatCount=0,
downTime=99603988000000, eventTime=99603988000000
```

Anexo 10

Pantalla de inicio de sesión



Inicie sesión

Email
jhondoe@gmail.com

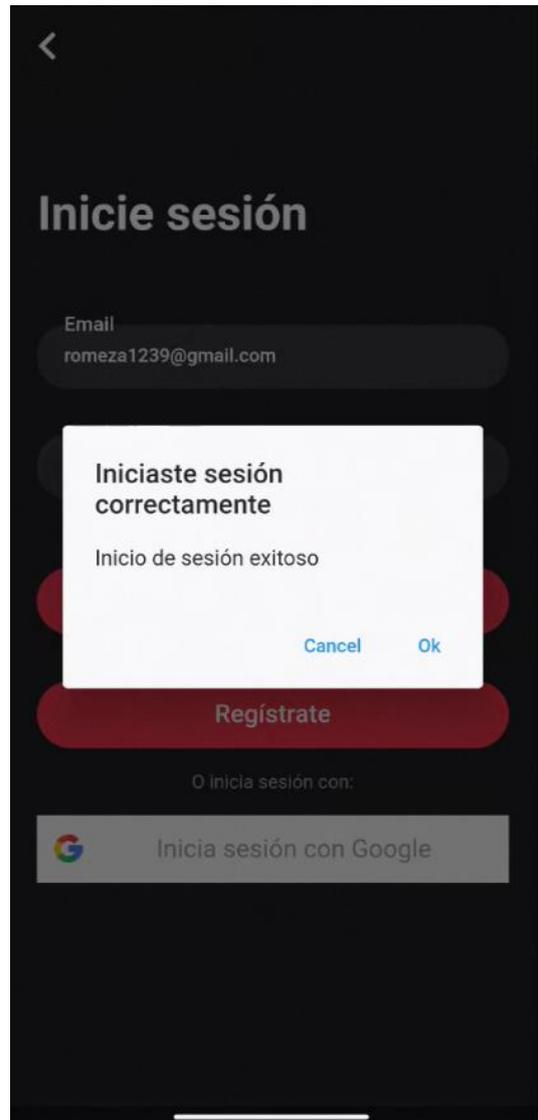
Contraseña

Iniciar sesión

Regístrate

O inicia sesión con:

 Inicia sesión con Google



Inicie sesión

Email
romeza1239@gmail.com

Iniciaste sesión correctamente

Inicio de sesión exitoso

[Cancel](#) [Ok](#)

Regístrate

O inicia sesión con:

 Inicia sesión con Google

Anexo 11

Pantalla de registro de usuario

<

Regístrate

Nombres
Miguel

Apellidos
Hidalgo

Email
mhidalgo@gmail.com

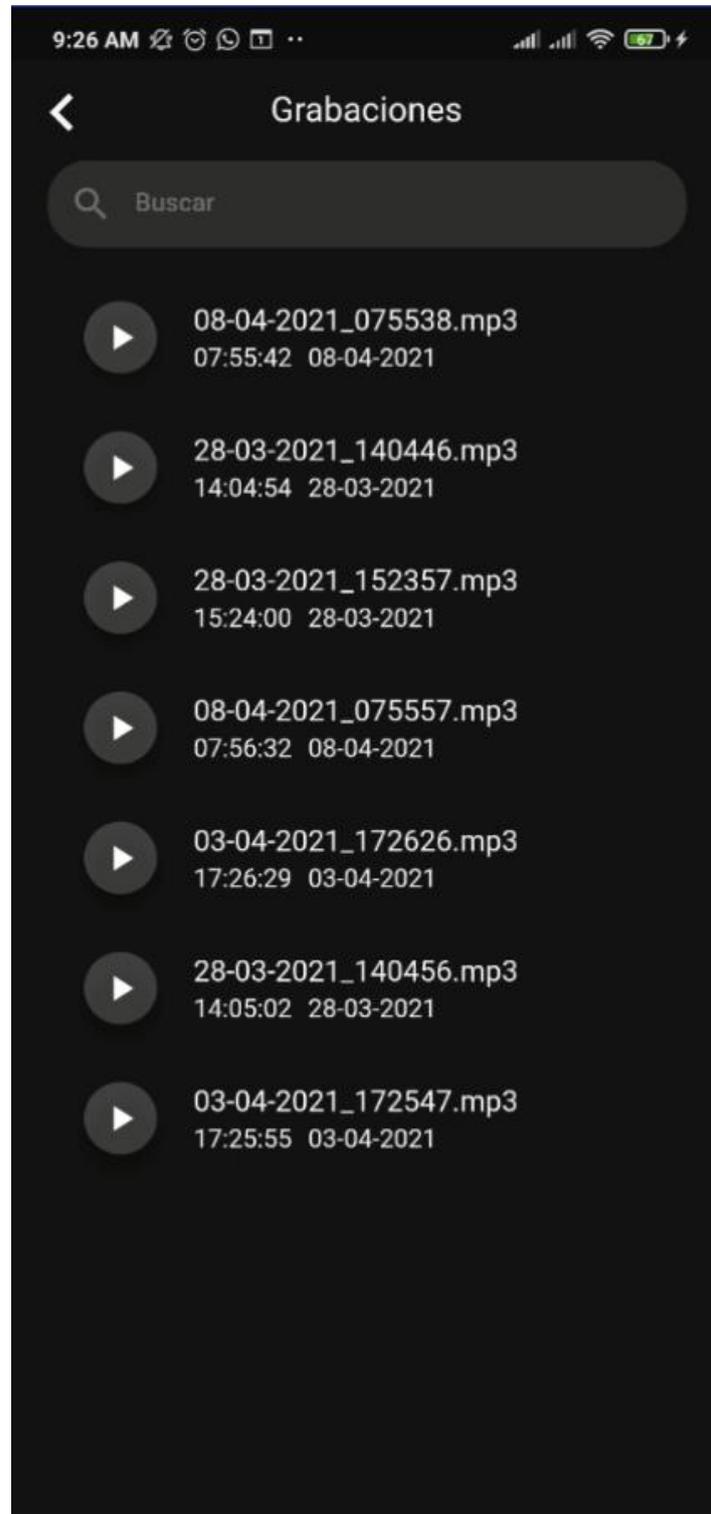
Contraseña
.....

Confirmar contraseña
.....

Regístrate

Anexo 12

Vista de listado de grabaciones registradas



Anexo 13

Matriz de resultados del proyecto de investigación

Problema	Objetivos	Resultado	Actividades y tareas	Método/Procedimiento/Técnica	Indicadores de resultado general	Instrumento de evaluación
¿Cómo identificar la violencia doméstica psicológica por voz al momento que el incidente esté en curso en el distrito de El tambo Huancayo Junín en el periodo del 2021?	Objetivo general Elaborar un sistema de reconocimiento de violencia doméstica psicológica por voz en el distrito de El Tambo, Huancayo en el 2021.	Resultado general Tener un sistema de reconocimiento de violencia doméstica psicológica por voz funcional y sencilla de usar, probada en el distrito de Huancayo en el 2021	-	-	-	
	Objetivos específicos Identificar los factores pertinentes en un sistema que identifique agresiones verbales.	Resultados específicos Tecnologías y requerimientos del sistema identificador de agresiones verbales definidas para llevar a cabo el desarrollo e implementación del sistema.	Lista de requisitos	Historias de usuario	<i>Backlog</i> del proyecto	Backlog y tabla de criterios de aceptación por tarea
	Objetivos específicos: Identificar las características lingüísticas de la	Resultados específicos: Inventario fraseológico de las frases violentas, clasificar los tipos de	Plan del proyecto	Estimación por puntos de historia (póker y planificación)	Planificación de <i>Sprint</i> y puntuación de historias	Cronograma y <i>sprints backlogs</i>
			Creación de inventario fraseológico adaptado	Artículo fraseológico de los insultos	Lista de frases y su clasificación	Índice de coeficiente de validación de inventario fraseológico

	violencia psicológica familiar.	frases violentas existentes.	Validación de inventario fraseológico	Uso de coeficiente de validez de contenido.	Coeficiente de validación de inventario fraseológico	Cuestionario de validación de contenido
	Objetivos específicos: Formular el diseño del sistema de procesamiento de lenguaje natural capaz de reconocer frases ofensivas y/o violentas.	Resultados específicos: Esquema, flujo y arquitectura del procesamiento clasificador del sistema identificador de violencia familiar	Diseño de interfaz gráfica	Principio UX investigación de usuarios	Estadística recolectada por el MIMP e IPSOS	Informes estadísticos de uso de celulares
	Objetivos específicos: Medir la eficacia del modelo de procesamiento de lenguaje natural capaz de reconocer frases ofensivas y violentas	Resultados específicos: Instrumento cuantizador de confianza para el modelo generado, pruebas de usabilidad del sistema de reconocimiento	Pruebas de fiabilidad del sistema de reconocimiento de violencia verbal	Pruebas de funcionalidad de modelo clasificador de violencia verbal	Exactitud, Especificidad, Tasa de error y sensibilidad	Matriz de confusión
			Pruebas de usabilidad del sistema de reconocimiento de violencia	Evaluación de usabilidad	Índice de usabilidad	Encuesta de usabilidad

Anexo 14

Pregunta de tiempo de respuesta del formulario

¿Cuánto fue el tiempo de respuesta en que empezó a grabar después de dar una frase de insulto?

- menos de 1 segundo
- 1 a 2 segundos
- 3 a 4 segundos
- más de 4 segundos
- nunca respondió

Anexo 15

Formulario de escala de usabilidad usada con google Form

Cuestionario app Painless *

	1 Totalmente desacuerdo	2 en desacuerdo	3 un poco de acuerdo	4 de acuerdo	5 Totalmente de acuerdo
Pienso que me gustaría usar este sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontré al sistema innecesariamente complejo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pienso que el sistema fue muy sencillo de usar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sentí que necesitaba una persona de soporte técnico para poder usar el sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontré que las diversas funciones en este sistema bien integradas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Opino que hubo demasiada inconsistencia en el sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Imagino que la mayoría de las personas aprenderían a utilizar este sistema rápidamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Me sentí algo incómodo al utilizar este sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Me sentí muy seguro al utilizar este sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Anexo 16

Tabla de resultados de escala de usabilidad

Nombres	Dirección de correo electrónico	N° Pregunta										Escala de usabilidad
		1	2	3	4	5	6	7	8	9	10	
Carlos Egoávil	ceev95@gmail.com	5	1	5	1	5	1	5	1	5	1	100
Claudia Villarubia	villarrubia.pacheco@gmail.com	5	1	5	1	5	1	5	1	5	1	100
Giovany Segura	itari010@gmail.com	5	1	5	1	4	1	5	1	5	1	97.5
Kris Meza	47979565@continental.edu.pe	5	1	5	5	5	1	5	1	5	1	90
Claudia Cuesta	michiclau7@gmail.com	5	3	4	4	4	1	4	1	5	2	77.5
Paola Ninanya	fcpaen@gmail.com	5	1	4	2	4	1	4	1	4	2	85
Naysha Ramos	naysharamos1@gmail.com	5	2	5	1	5	1	5	1	4	1	95
Naysha Meza	mezanaysha237@gmail.com	5	4	4	2	5	2	5	4	5	2	75
Nicole Segura	nicolegasm@gmail.com	5	1	5	1	5	1	4	1	5	2	95
JANINA CUBA URIBE	72659369@continental.edu.pe	2	1	2	1	2	1	2	1	2	1	62.5
Alejandra Paucar	aleepaucarpichiule@gmail.com	5	1	4	4	5	1	5	1	5	3	85
Victoria Barrios	victoriabarriosolano15@gmail.com	4	1	5	1	5	2	5	1	5	2	92.5
Maria Traverso	mtraversopaitamala@gmail.com	5	2	5	1	5	1	5	1	5	1	97.5
Gabriela Montalvo	gabrielamms1999@gmail.com	4	2	5	2	4	1	5	2	4	3	80
Virna Cueva	virnacueva98@gmail.com	5	1	5	2	5	1	5	1	4	1	95
Promedio											88.5	

Anexo 17

Tabla de coeficiente de validez de contenido

Ítem	Juez 01	Juez 02	Juez 03	Juez 04	Juez 05	Sx _i	M _x	CVC _i	P _{ei}	CVC _{ic}
Ítem 1	1	3	3	5	2	14	2.8	0.56	0.00032	0.55968
Ítem 2	5	5	4	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 3	5	5	4	4	4	22	4.4	0.88	0.00032	0.87968
Ítem 4	4	2	3	3	4	16	3.2	0.64	0.00032	0.63968
Ítem 5	5	5	4	3	3	20	4	0.8	0.00032	0.79968
Ítem 6	4	5	5	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 7	5	4	5	3	3	20	4	0.8	0.00032	0.79968
Ítem 8	3	3	4	2	2	14	2.8	0.56	0.00032	0.55968
Ítem 9	5	4	4	4	2	19	3.8	0.76	0.00032	0.75968
Ítem 10	5	5	4	5	4	23	4.6	0.92	0.00032	0.91968
Ítem 11	2	5	4	5	4	20	4	0.8	0.00032	0.79968
Ítem 12	3	5	4	5	3	20	4	0.8	0.00032	0.79968
Ítem 13	3	5	1	3	2	14	2.8	0.56	0.00032	0.55968
Ítem 14	3	4	2	2	1	12	2.4	0.48	0.00032	0.47968
Ítem 15	5	3	3	2	2	15	3	0.6	0.00032	0.59968
Ítem 16	4	4	5	2	1	16	3.2	0.64	0.00032	0.63968
Ítem 17	5	5	5	3	3	21	4.2	0.84	0.00032	0.83968
Ítem 18	5	5	5	2	2	19	3.8	0.76	0.00032	0.75968
Ítem 19	5	5	5	2	2	19	3.8	0.76	0.00032	0.75968
Ítem 20	5	5	4	4	1	19	3.8	0.76	0.00032	0.75968
Ítem 21	5	5	5	4	1	20	4	0.8	0.00032	0.79968
Ítem 22	3	4	5	2	4	18	3.6	0.72	0.00032	0.71968
Ítem 23	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 24	5	5	5	3	2	20	4	0.8	0.00032	0.79968
Ítem 25	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Ítem 26	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968
Ítem 27	5	5	5	2	3	20	4	0.8	0.00032	0.79968
Ítem 28	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 29	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968
Ítem 30	5	5	4	2	2	18	3.6	0.72	0.00032	0.71968
Ítem 31	5	2	5	3	3	18	3.6	0.72	0.00032	0.71968
Ítem 32	5	4	4	2	2	17	3.4	0.68	0.00032	0.67968
Ítem 33	5	3	5	3	4	20	4	0.8	0.00032	0.79968
Ítem 34	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Ítem 35	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 36	5	4	5	4	4	22	4.4	0.88	0.00032	0.87968
Ítem 37	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968
Ítem 38	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 39	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 40	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 41	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 42	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 43	5	5	5	4	4	23	4.6	0.92	0.00032	0.91968
Ítem 44	5	3	5	3	4	20	4	0.8	0.00032	0.79968
Ítem 45	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 46	5	5	5	2	4	21	4.2	0.84	0.00032	0.83968
Ítem 47	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 48	5	4	5	2	4	20	4	0.8	0.00032	0.79968
Ítem 49	5	3	5	2	4	19	3.8	0.76	0.00032	0.75968
Ítem 50	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968
Ítem 51	5	3	5	2	4	19	3.8	0.76	0.00032	0.75968
Ítem 52	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968

Ítem 53	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968	
Ítem 54	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968	
Ítem 55	4	3	5	3	4	19	3.8	0.76	0.00032	0.75968	
Ítem 56	2	3	5	3	4	17	3.4	0.68	0.00032	0.67968	
Ítem 57	3	3	5	5	4	20	4	0.8	0.00032	0.79968	
Ítem 58	3	4	5	4	4	20	4	0.8	0.00032	0.79968	
Ítem 59	5	4	5	4	4	22	4.4	0.88	0.00032	0.87968	
Ítem 60	5	4	5	2	4	20	4	0.8	0.00032	0.79968	
Ítem 61	5	3	5	3	4	20	4	0.8	0.00032	0.79968	
Ítem 62	4	3	5	4	4	20	4	0.8	0.00032	0.79968	
Ítem 63	5	4	5	2	4	20	4	0.8	0.00032	0.79968	
Ítem 64	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968	
Ítem 65	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968	
Ítem 66	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968	
Ítem 67	3	3	4	5	4	19	3.8	0.76	0.00032	0.75968	
Ítem 68	5	3	3	2	4	17	3.4	0.68	0.00032	0.67968	
Ítem 69	4	2	5	2	4	17	3.4	0.68	0.00032	0.67968	
Ítem 70	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
Ítem 71	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
Ítem 72	5	3	5	5	4	22	4.4	0.88	0.00032	0.87968	
Ítem 73	5	5	5	5	4	24	4.8	0.96	0.00032	0.95968	
Ítem 74	5	5	5	3	4	22	4.4	0.88	0.00032	0.87968	
Ítem 75	3	3	5	4	4	19	3.8	0.76	0.00032	0.75968	
Ítem 76	4	3	5	2	4	18	3.6	0.72	0.00032	0.71968	
Ítem 77	5	2	5	3	4	19	3.8	0.76	0.00032	0.75968	
Ítem 78	5	4	5	3	4	21	4.2	0.84	0.00032	0.83968	
Ítem 79	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
Ítem 80	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
Ítem 81	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
Ítem 82	5	4	5	5	4	23	4.6	0.92	0.00032	0.91968	
										CVC	0.810411707