

Universidad Autónoma Metropolitana  
Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación

Reporte Final de Proyecto de Integración  
Chatbot con análisis de sentimientos para fines de mercadotecnia

Modalidad: Proyecto Tecnológico

Trimestre 2022 Invierno

Juan Pablo Delgado Frías  
2172001260  
al2172001260@azc.uam.mx

Asesora  
Dra. Ángeles Belém Priego Sánchez  
Profesora Asociada  
Departamento de Sistemas  
abps@azc.uam.mx

Co-asesor  
Dr. David Eduardo Pinto Avendaño  
Profesor Titular  
Benemérita Universidad Autónoma de Puebla  
dpinto@cs.buap.mx

26 de mayo de 2022

## **Declaratoria**

Yo, Dra. Ángeles Belém Priego Sánchez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Dra. Ángeles Belém Priego Sánchez

Yo, Dr. David Eduardo Pinto Avendaño, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Juan Pablo Delgado Frías, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Juan Pablo Delgado Frías

## Resumen

En la actualidad existen numerosas empresas y organizaciones que usan diferentes técnicas para analizar grandes conjuntos de datos con diferentes fines, entre una de las fuentes de información más importantes a analizar se encuentran las reseñas u opiniones de usuarios o clientes sobre un bien o servicio.

Existen diversas formas de recopilar este tipo de información, ya sea de forma verbal o escritas física o electrónicamente, aunque para el análisis posterior de la información obtenida puede ser más recomendable trabajar con datos obtenidos electrónicamente, ya que se obtiene facilidad para trabajar con grandes cantidades de datos de manera más eficiente haciendo uso de funciones computacionales.

La principal problemática que se puede presentar en la obtención de información de este tipo radica en la forma de administrarla, ya que se pueden presentar grandes cantidades de datos llegando incluso en un mismo instante, lo que haría laboriosa su administración si no se tiene una buena estructura predefinida para almacenarla.

El presente proyecto logra brindar una alternativa para resolver dicha problemática, que consiste en una página electrónica capaz de recopilar datos relevantes de usuarios o clientes, tales como el nombre, la edad, género y opiniones de estos mismos, y almacenarlos automáticamente en una estructura de datos.

La forma de hacerlo es mediante un formulario de registro desarrollado para que sea lo primero que aparezca en la página web, mismo que al completarse, permite el acceso a un servicio parecido al de una mensajería instantánea, diseñado para que el usuario pueda ingresar sus comentarios u opiniones, y al ser enviados, reciban una respuesta automatizada al mismo tiempo que su comentario es clasificado de acuerdo al sentimiento detectado, y posteriormente sea almacenado junto con los datos de registro en una base de datos diseñada con una estructura especial que permita un análisis posterior a estos datos.

El aporte principal de este de este proyecto radica en la nueva información generada tras emplear un método de agrupación de los datos finales recopilados que permite identificar patrones, mismos que pueden ser aprovechados por las empresas u organizaciones para incrementar el éxito de sus decisiones tomadas.

# Contenido

1. Introducción .....	1
2. Antecedentes .....	2
2.1. Referencias internas.....	2
2.2. Referencias externas.....	3
3. Justificación .....	5
4. Objetivos .....	6
4.1. Objetivo General .....	6
4.2. Objetivos Específicos.....	6
5. Marco Teórico .....	7
5.1. Chatbot.....	7
5.2. Tipos de chatbot.....	7
5.3. Inteligencia artificial .....	8
5.4. Minería de datos .....	8
5.5. Técnicas de minería de datos.....	8
5.6. Machine Learning.....	9
5.7. Lenguaje natural.....	9
5.8. Procesamiento de Lenguaje natural .....	10
5.9. Análisis de sentimientos.....	10
5.10. Biblioteca.....	10
5.11. Python .....	10
5.12. Chatterbot .....	10
5.13. Flask.....	11
5.14. Sentiment_Analysis_Spanish .....	11
5.15. HTML .....	11
5.16. CSS.....	11
5.17. JavaScript.....	11
5.18. JQuery .....	11
5.19. WEKA.....	11
6. Desarrollo del proyecto.....	12
6.1. Módulo de chat .....	12
6.2. Módulo de análisis de sentimientos .....	20
6.3. Módulo de clasificación de opiniones.....	21

6.4. Base de datos .....	22
7. Resultados .....	26
8. Análisis y discusión de resultados .....	29
8.1. Resultados relacionados con el análisis de sentimientos .....	29
8.2. Resultados relacionados con la minería de datos .....	31
9. Conclusiones.....	38
10. Referencias.....	39
11. Apéndices .....	41
A. Script SQL de la base de datos .....	41
B. Script de Chatbot .....	43
C. Página con HTML.....	46
D. Hoja de estilo en cascada (CSS) .....	47
E. JavaScript de manejo del ChatBot.....	50

## Índice de tablas

Tabla 1: Comparación cualitativa de los trabajos relacionados con el proyecto terminal realizado ...	3
Tabla 2: Clasificación de un mensaje .....	22
Tabla 3: Valores por defecto de la tabla sexo .....	23
Tabla 4: Valores por defecto de la tabla sentimiento .....	24

## Índice de figuras

Figura 1: Diagrama de bloques sobre el proyecto .....	12
Figura 2: Diagrama de bloques sobre el módulo de chat.....	12
Figura 3: Diagrama sobre modelo de programación MVC.....	19
Figura 4: Diagrama de bloques sobre el módulo de análisis de sentimientos.....	20
Figura 5: Modelo relacional de la BD .....	22
Figura 6: Clasificación de mensaje muy negativa.....	26
Figura 7: Clasificación de mensaje negativa.....	27
Figura 8: Clasificación de mensaje neutra.....	27
Figura 9: Clasificación de mensaje positiva.....	28
Figura 10: Clasificación de mensaje muy positiva.....	28
Figura 11: Comentario muy negativo.....	29
Figura 12: Comentario negativo.....	30
Figura 13: Clasificación errónea de mensaje.....	31

Figura 14: Minería de datos (50 registros) .....	32
Figura 15: Minería de datos (100 registros) .....	34
Figura 16: Minería de datos (200 registros) .....	35

## Índice de gráficas

Gráfica 1: Análisis de 50 registros .....	32
Gráfica 2: Análisis de 100 registros .....	34
Gráfica 3: Análisis de 200 registros .....	36

## Índice de códigos

Código 1: Instalación de la biblioteca chatterbot .....	13
Código 2: Instalación de la biblioteca flask .....	13
Código 3: Instalación de la biblioteca jinja2 .....	13
Código 4: Instalación de la biblioteca nltk .....	13
Código 5: Instalación de la biblioteca sqlalchemy .....	14
Código 6: Inicialización de flask .....	14
Código 7: Método de ejecución de flask .....	14
Código 8: Inicialización del chatbot .....	15
Código 9: Inicialización de lista de entrenamiento del chatbot .....	15
Código 10: Entrenamiento del chatbot .....	15
Código 11: Método para cargar HTML .....	16
Código 12: Método para obtener respuestas .....	16
Código 13: Enlace a archivo CSS .....	17
Código 14: Enlace a archivo JavaScript .....	17
Código 15: Hoja de estilos del chatbot .....	17
Código 16: Script para almacenar respuestas .....	18
Código 17: Enlace de JQuery .....	19
Código 18: Instalación de la biblioteca de análisis de sentimientos .....	20
Código 19: Inicialización del análisis de sentimientos .....	21
Código 20: Obtención de puntaje del análisis de sentimientos .....	21
Código 21: Método para insertar en BD .....	25
Código 22: Inicialización de objeto DataBase .....	25

## **1. Introducción**

Hoy en día, muchas personas realizan diversas actividades en internet por medio de ordenadores o dispositivos móviles, y dentro de éstas, nos encontramos con actividades que se han visto en aumento debido a la situación sanitaria que se atraviesa desde 2020, tales como: compras en línea, suscripciones, o contrataciones de servicios. Este tipo de actividades convierten al usuario en un cliente que obtiene un bien o servicio del cual se puede sentir satisfecho o no, por lo que es importante que se consideren sus opiniones para así poder ofrecer mejores bienes o servicios, sin embargo, esto puede representar una problemática dada la forma en que se puede capturar y manejar la información.

Una forma popular actualmente de manejar este tipo de situación es mediante el uso de chatbots [1] en páginas web, por lo que en este proyecto se implementó esta solución, pero añadiendo funcionalidades extra, como la de evaluar las opiniones de los clientes y guardar información vital de estas opiniones que se pueda analizar para obtener nueva información relevante sobre un producto o servicio. Las acciones mencionadas anteriormente son posibles gracias al uso de áreas del Procesamiento de Lenguaje Natural (PLN) conocidas como análisis de sentimientos y clasificación de polaridad de textos [2], así como el uso de minería de datos [3].

## 2. Antecedentes

### 2.1. Referencias internas

#### ■ **Minería de opiniones sobre textos en Twitter [4]**

El objetivo de este proyecto es el de desarrollar un sistema que permita la recolección de comentarios escritos en la red social Twitter para después poder analizarlos y así obtener información valiosa sobre las tendencias que se muestran sobre diversos temas. La recolección se basa en una palabra o frase que debe ser introducida previamente por el usuario, y de acuerdo con esto se realiza un filtrado de los comentarios en la red social.

#### ■ **Sistema para la clasificación de opiniones generadas en Twitter usando redes neuronales artificiales [5]**

El objetivo de este proyecto es el de realizar una clasificación de opiniones recopiladas de la red social Twitter de acuerdo con la polaridad que dicho comentario presenta (positivo, negativo y neutro), esto último es logrado mediante técnicas de procesamiento de lenguaje natural y el uso de redes neuronales artificiales. Finalmente, para maximizar la certeza de un clasificado correcto de opiniones casi en su totalidad, se hace uso de la herramienta WEKA para poder realizar un análisis sobre los datos obtenidos, dando así como resultado información útil que muestra tendencias claras sobre servicios, productos o diversos temas de conversación.

#### ■ **Algoritmos de aprendizaje automático para el análisis de opiniones a partir de textos en español [6]**

Este proyecto terminal consiste en la implementación de cuatro algoritmos relacionados a minería de textos con el fin de comparar las diferencias que hay entre realizar este tipo de procesos en español y en otros idiomas. Los algoritmos implementados en este proyecto son los siguientes: K-vecinos más cercanos (KNN), Bayes Naïve, J48, Maquinas de Soporte Vectorial (SVM).

## 2.2. Referencias externas

### ■ **Detección de estados de ánimo mediante sentiment analysis en hispanohablantes** [7]

Esta tesis muestra la implementación de un modelo computacional con el fin de detectar el estado de ánimo de personas hispanohablantes empleando técnicas de reconocimiento de palabras clave y análisis de sentimientos. Los textos de los cuales se pretende detectar el estado de ánimo son extraídos mediante una conversación de un chat automatizado en celular que incita al usuario a dar detalles sobre su estado de ánimo

### ■ **Aplicación del Análisis de Sentimientos a la Evaluación de Datos Generados en Medios Sociales** [8]

Esta tesis doctoral propone analizar el impacto y la eficiencia del análisis de sentimientos de textos recopilados en medios sociales. La tesis incluye un estudio sobre la confiabilidad de su algoritmo utilizado para el análisis de sentimientos, además de concluir que el tema de minería de datos podría resultar muy beneficioso para la toma de decisiones de compañías e individuos aprovechando la cantidad masiva de opiniones que hay cada día en medios sociales.

### ■ **Identificación de depresión mediante el análisis de sentimientos** [9]

Este proyecto consiste en el desarrollo de una aplicación móvil que realiza preguntas puntuales a un usuario, y las respuestas introducidas serán analizadas mediante técnicas de procesamiento de lenguaje natural, siendo más específicos, mediante análisis de sentimientos, permitiendo así reconocer casos posibles de depresión. La aplicación sólo pretende ser un apoyo en el tema de detección de la depresión y no reemplaza a un especialista.

La Tabla 1 muestra una comparación de los trabajos relacionados con el proyecto terminal realizado.

Referencia	Similitudes	Diferencias
[4]	Ambos proyectos están enfocados en la minería de opiniones con el fin de producir información útil acerca de las tendencias que se muestran respecto a cada opinión.	El método de recolección de información es distinto, ya que en el proyecto referenciado se obtienen las opiniones directo de una red social (Twitter), mientras

		que en este proyecto se propone hacerlo mediante un chat automatizado que realice una clasificación de las opiniones.
[5]	En ambos proyectos se realiza una clasificación por medio de algoritmos relacionados con el análisis de sentimientos, además de usar el software WEKA para realizar un análisis sobre los datos obtenidos	Las opiniones se recolectan de una manera distinta, mientras que en el proyecto referenciado se obtienen las opiniones directo de una red social (Twitter), en este proyecto las opiniones se obtienen mediante un chat automatizado.
[6]	Ambos proyectos utilizan la minería de textos en español	El proyecto referenciado está enfocado en una comparativa de diferentes algoritmos, mientras que en este proyecto se busca la producción de información relevante para mercadotecnia, además de que la forma de recolectar opiniones es distinta.
[7]	Los proyectos buscan realizar una clasificación de textos ingresados por un usuario mediante un chat automatizado.	El proyecto referenciado sólo busca realizar una predicción sobre el estado de ánimo de un texto ingresado. El proyecto propuesto en este documento pretende realizar un análisis de acuerdo con la clasificación de sentimiento asignado a cada texto u opinión.
[8]	Se utiliza el análisis de sentimientos para la minería de datos en textos en español con el fin de obtener información valiosa que permita apreciar tendencias claras en un conjunto de opiniones que faciliten a empresas u organizaciones planear mejores decisiones.	La tesis doctoral está enfocada en recopilar datos de medios sociales, mientras que en este proyecto se propone hacerlo mediante un chat automatizado que realice una clasificación de las opiniones.
[9]	Se utilizan técnicas de análisis de sentimientos con el fin de proporcionar información relevante al usuario.	El proyecto referenciado está más enfocado en el sector médico o psicológico, además de enfocarse en las entradas de texto de un solo usuario mediante el uso de una aplicación móvil.

**Tabla 1.** Comparación cualitativa de los trabajos relacionados con el proyecto terminal realizado

### **3. Justificación**

La implementación de este modelo de chatbot es importante porque ayuda a aplicar la filosofía del término de producción ajustada [10], que significa identificar lo que tiene valor para el consumidor, y de esta manera optimizar el proceso de producción y así eliminar o minimizar desperdicios, logrando así una disminución en el impacto que se tiene hacia el medio ambiente, y no sólo eso, también haciendo más eficiente la toma de decisiones de empresas u organizaciones basada en información recopilada de los mismos clientes.

Otro punto importante es la disminución de costos para las empresas u organizaciones, ya que, de optar por un sistema más tradicional, se tendrían que contratar personas específicamente para la recopilación de las opiniones, así como el análisis de las opiniones recopiladas. Por otro lado, la solución que brinda este proyecto es un sistema automatizado que puede analizar sentimientos en las oraciones obtenidas por medio de un chat en una página web, lo que también brinda como otro punto positivo, un servicio continuo las 24 horas del día, en el que el cliente puede responder sin presiones de tiempo y desde la comodidad de su hogar, lo que es un factor muy importante hoy en día dada la situación sanitaria por la que se está atravesando, y con lo que seguramente los clientes se sienten más cómodos.

## **4. Objetivos**

### **4.1. Objetivo General**

Implementar un chatbot automatizado para fines de mercadotecnia mediante el uso de análisis de sentimientos en opiniones de productos o servicios de una empresa u organización.

### **4.2. Objetivos Específicos**

1. Desarrollar un módulo de chat capaz de adquirir opiniones de clientes.
2. Desarrollar un módulo de análisis de sentimientos que asigne una ponderación numérica de acuerdo con el sentimiento detectado en una opinión.
3. Desarrollar un módulo de clasificación de opiniones que almacene la información obtenida de cada opinión indicada con el sentimiento detectado.

## **5. Marco Teórico**

### **5.1. Chatbot**

Ya que el proyecto desarrollado en esencia es un Chatbot, es de vital importancia conocer su concepto. En [1] se define un chatbot como un programa informático que permite a los humanos interactuar con dispositivos digitales (de forma escrita o hablada) simulando como si se comunicaran con una persona real.

### **5.2. Tipos de chatbot**

En [11] se detallan los diferentes tipos de chatbot, los cuales pueden variar de acuerdo con el uso que se le quiera dar:

- De acuerdo con su inteligencia:
  - Chatbots tontos: Son la opción más básica de chatbots, están programados de manera que responden de una forma rígida siguiendo siempre el mismo orden en la conversación.
  - Enfocados en palabras: Son configurados para responder a palabras clave, sin embargo, siguen manteniendo cierto grado de rigidez.
  - Con inteligencia artificial: Son la opción más avanzada y actual, sus algoritmos son capaces de interpretar el lenguaje humano y responder de una manera acorde haciéndolos casi parecer como una conversación con otra persona.
- De acuerdo con el tipo de interacción:
  - Textuales: Son la opción básica de interacción, sólo son capaces de comunicarse con los usuarios a través de mensajes escritos.
  - Multimedia: Además de incluir comunicación a través de mensajes escritos, también es capaz de comunicarse mediante imágenes, vídeos, GIFs, entre otros.
  - De voz: Son una opción más moderna capaz de interpretar peticiones habladas de un usuario y responder a través de un altavoz.
- De acuerdo con el canal o medio usado para su uso:

- Sitio web: Son la opción más conocida por los usuarios, son implementados en páginas web, usualmente para servicio al cliente.
- Redes sociales: Su principal función es la de enganchar usuarios de una red social mediante respuestas instantáneas las 24 horas del día, son especialmente útiles para brindar información.
- Mensajería instantánea: Su principal función es la de cubrir servicio al cliente, pero enfocado en aplicaciones móviles.

De acuerdo con las clasificaciones listadas anteriormente, el chatbot que se desarrolla en este proyecto abarca tres clasificaciones diferentes: con inteligencia artificial, textual y en sitio web.

### **5.3. Inteligencia artificial**

En [12] se define a la inteligencia artificial como un sistema o máquina que imita a la inteligencia humana que puede realizar tareas y mejorarlas por medio de recolecciones de datos repetidas varias veces.

### **5.4. Minería de datos**

El término de minería de datos se refiere al uso de técnicas y tecnologías que permiten analizar grandes conjuntos de datos almacenados, ya sea de forma automática o semiautomática. Su objetivo consiste en encontrar patrones repetitivos que expliquen el comportamiento de los datos analizados, por lo tanto, los datos son el medio o la base para llegar a conclusiones y transformar estos datos en información [13].

### **5.5. Técnicas de minería de datos**

En [13], se muestra un listado de las técnicas con las que se puede llevar a cabo un minado de datos:

- Asociación: Es una de las técnicas más usadas, también se le conoce como “técnica de relación” ya que relaciona elementos para identificar patrones. Un ejemplo de esta técnica es el análisis de artículos de una cesta de compra de un supermercado, en la que se pretende conocer todos los productos que los clientes compran juntos habitualmente.

- Agrupación o *clustering*: Esta técnica consiste en agrupar datos significativos que comparten las mismas características. La agrupación suele confundirse con la clasificación, pero su diferencia radica en que la agrupación coloca datos en clases predefinidas. Es importante resaltar que esta es la técnica usada en este proyecto terminal.
- Clasificación: Esta técnica clasifica elementos o variables en un conjunto de datos. Para lograr la clasificación se utiliza programación lineal, estadísticas, árboles de decisiones y redes neuronales artificiales, sólo por mencionar algunas.
- Predicción: Como su nombre lo sugiere, realiza una predicción sobre la relación que existe entre variables independientes y dependientes. Para lograr que una predicción sea más acertada y acorde a la realidad, se requiere tomar en cuenta la mayoría de los factores que afectan a dicha información.
- Patrones secuenciales: Esta técnica funciona de manera parecida a la predicción, ya que usa datos de diferentes periodos, sin embargo, no busca realizar una predicción, sino usar los patrones encontrados en la información o datos históricos sobre el tema.

## **5.6. Machine Learning**

Es uno de los campos de la inteligencia artificial, su principal propósito es el de dotar con algoritmos a los ordenadores para así brindarles la capacidad de identificar patrones en datos masivos y elaborar predicciones de forma autónoma, lo que permite funcionar a estos sistemas sin necesidad de estar siendo programados [14].

## **5.7. Lenguaje natural**

En este proyecto se trabaja con lenguaje natural y una definición apegada a lo que se trabaja en este proyecto es la encontrada en [15] donde se define al lenguaje natural como “lengua que usan los individuos para interactuar a través de alguna forma de comunicación sea escrita, oral o no verbal. Una base para comprender el lenguaje natural es el idioma que empleamos cuando hablamos (castellano, inglés, chino, entre otros).”

## **5.8. Procesamiento de lenguaje natural**

El procesamiento de lenguaje natural es otro de los campos de la inteligencia artificial, su principal propósito es el de investigar la manera de comunicar máquinas con personas mediante el uso de lenguas naturales como el español o cualquier otro idioma [2].

## **5.9. Análisis de sentimientos**

Se refiere a los diferentes métodos de lingüística computacional que ayudan a identificar y extraer información existente en el mundo digital (foros, webs, etc.) [16]. En otras palabras, el análisis de sentimientos se usa como un método automatizado para clasificar en función de una connotación positiva o negativa de un documentos o texto.

## **5.10. Biblioteca**

En [17] se define a una biblioteca como “conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular.”

## **5.11. Python**

Lenguaje de programación de alto nivel que de acuerdo con la página oficial [18] permite trabajar de manera rápida e integrar sistemas más eficientemente.

Python cuenta con una gran cantidad de bibliotecas relacionadas a la inteligencia artificial, las cuales son accesibles mediante el manejador de paquetes “pip”, por lo que instalar bibliotecas en Python es algo muy sencillo, ya que sólo se requiere escribir un comando. La sintaxis del comando es la siguiente: `pip install <nombre de la biblioteca>`.

## **5.12. Chatterbot**

Biblioteca de Python que usa algoritmos del campo de la inteligencia artificial para producir diferentes tipos de respuestas automatizadas que simulan una conversación [19], cuenta con diferentes formas de personalización para hacer la interacción lo más parecida a hablar con un humano.

### **5.13. Flask**

Biblioteca escrita en Python concebida para facilitar el desarrollo de aplicaciones web [20], es especialmente útil para realizar pruebas de proyectos pequeños.

### **5.14. Sentiment\_Analysis\_Spanish**

Biblioteca de Python que usa métodos estadísticos para predecir el sentimiento de oraciones en español. Para funcionar de una manera más certera fue entrenada usando más de 800000 opiniones de usuarios de diferentes páginas web que ofrecen servicios por internet [21].

### **5.15. HTML**

Lenguaje que define el significado y estructura del contenido web mediante marcas, suele ser usado junto con documentos de descripción de apariencia y/o funcionalidad [22].

### **5.16. CSS**

Es un lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML, indica al ordenador la forma de renderizado de cada elemento para así estilizar y mostrar en pantalla [23].

### **5.17. JavaScript**

En [24] se define JavaScript como “lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.”

### **5.18. JQuery**

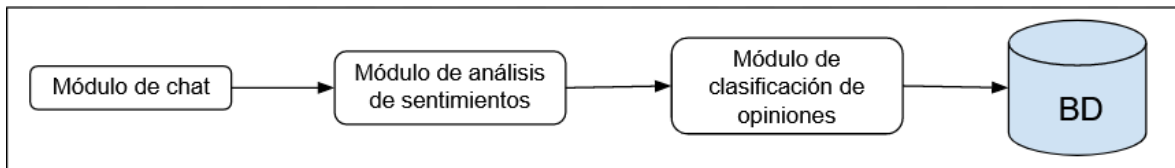
Biblioteca de JavaScript que permite manejar documentos HTML de una manera más simple gracias a sus funciones para manipular eventos, animaciones y documentos [25].

### **5.19. WEKA**

En [26] se define a WEKA como un *software* de código abierto que contiene una colección de algoritmos de machine learning para minería de datos, entre ellos, el usado en este proyecto terminal que es agrupación o clusters.

## 6. Desarrollo del proyecto

Para facilitar el entendimiento en la secuencia de pasos del proyecto, se realizó un diagrama de bloques, el cual se muestra en la Figura 1.

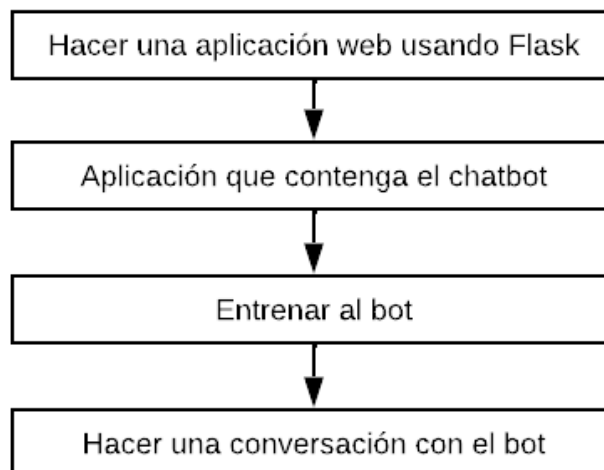


**Figura 1.** Diagrama de bloques sobre el proyecto.

La funcionalidad de cada uno de los módulos que componen al sistema se explica en las siguientes secciones:

### 6.1. Módulo de chat

Para comprender más fácilmente el desarrollo de este módulo se realizó un diagrama de bloques, mismo que se muestra en la Figura 2.



**Figura 2.** Diagrama de bloques sobre el módulo de chat.

La explicación y progreso de cada bloque de la Figura 2 se irá mencionando a lo largo de la siguiente explicación de cada código utilizado en este módulo:

- **Código en Python**

El código en Python (Apéndice B) requirió de diversas bibliotecas, por lo que es importante hablar de cada una y así resaltar la importancia de cada una dentro de este

código:

- Chatterbot: Esta es la biblioteca más importante en este código, ya que es la que maneja las respuestas automatizadas a los mensajes ingresados por un usuario.

Para su instalación, debe escribirse en consola el Código 1:

```
pip install chatterbot
```

**Código 1.** Instalación de la biblioteca chatterbot.

- Flask: Esta biblioteca facilita la creación de aplicaciones web, por lo que es muy importante para poder desplegar el chatbot en la página web donde se recolectan los comentarios ingresados por los usuarios.

Para su instalación, debe escribirse en consola el Código 2:

```
pip install flask
```

**Código 2.** Instalación de la biblioteca flask.

- Jinja2: Esta biblioteca es un motor de plantillas web, durante la realización del código no se hace uso de esta biblioteca directamente, pero es necesaria para el correcto funcionamiento de la biblioteca Flask.

Para su instalación, debe escribirse en consola el Código 3:

```
pip install jinja2
```

**Código 3.** Instalación de la biblioteca jinja2.

- Nltk: Esta biblioteca contiene funciones relacionadas con el lenguaje natural, durante la realización del código no se hace uso de esta biblioteca directamente, pero es necesaria para el correcto funcionamiento de la biblioteca Chatterbot.

Para su instalación, debe escribirse en consola el Código 4:

```
pip install nltk
```

**Código 4.** Instalación de la biblioteca nltk.

- SQLAlchemy: Esta biblioteca funciona como un controlador de bases de datos SQL, es usada por la biblioteca Chatterbot para almacenar de manera automática los mensajes que son recibidos por el bot de parte del usuario. Esta biblioteca en realidad no afecta el uso de la biblioteca Chatterbot, por lo que se puede considerar como opcional, sin embargo, el uso de la base de datos provee mejores resultados en las respuestas del bot, ya que la base de datos manejada funciona también para analizar qué respuestas obtuvieron un mayor éxito y cuáles no.

Para su instalación, debe escribirse en consola el Código 5:

```
pip install sqlalchemy
```

**Código 5.** Instalación de la biblioteca sqlalchemy.

Ahora que se ha cubierto la importancia e instalación de cada biblioteca en este código, se puede proseguir a hablar sobre el desarrollo del módulo.

El código de Python es el encargado de ejecutar una aplicación web, y con la descripción anterior sobre las bibliotecas, sabemos que esto se logra gracias a la biblioteca Flask, y dentro del código esto se logra mediante el Código 6:

```
app = Flask(__name__)
```

**Código 6.** Inicialización de flask.

El nombre app es una abreviación de *application*, y esto señala la creación de nuestra aplicación gracias a un método de Flask, sin embargo, esto sólo es el inicio y no realiza la ejecución de la aplicación por si solo, así que se debe agregar el Código 7:

```
If __name__ == "__main__":  
    app.run()
```

**Código 7.** Método de ejecución de flask.

De esta manera ahora sí se tiene una aplicación web con ayuda de Flask y que es ejecutable, esto por si sólo constituye el avance del primer bloque de la Figura 2.

Para el segundo bloque, que es la aplicación con el chatbot, tenemos ahora la

biblioteca Chatterbot, la cual cuenta con un método para crear un chatbot, el comando a ejecutar es el Código 8:

```
bot = ChatBot("ChatterBot",  
storage_adapter="chatterbot.storage.SQLStorageAdapter")
```

**Código 8.** Inicialización del chatbot.

En este método se especifica el nombre del chatbot, así como el tipo de almacenamiento a utilizar, en este caso se escoge un almacenamiento de las respuestas con SQL, mismo por el cual se utiliza la biblioteca SQLAlchemy que fue descrita anteriormente.

El tercer bloque es referente a entrenar el bot, y se consigue también con funciones incluidas en la biblioteca Chatterbot, tal como se muestra en el Código 9:

```
trainer = ListTrainer(bot)
```

**Código 9.** Inicialización de lista de entrenamiento del chatbot.

Esta instrucción prepara una lista vacía de sentencias lista para ser cargada con ejemplos de conversaciones, en este código, la estructura de las sentencias más importantes para el funcionamiento del bot son como las del Código 10:

```
trainer.train([  
    '00001',  
    'Comentario negativo',])
```

**Código 10.** Entrenamiento del chatbot.

Lo que implica esta instrucción es que al recibir un mensaje del tipo “00001” se contestará con la sentencia “Comentario negativo”. Sería imposible pensar que un usuario escribiera 00001 al bot, y la explicación de esto es que este mensaje está codificado y es resultado del módulo de clasificación de opiniones que se explicará más adelante. Existe una de estas instrucciones por cada sentimiento tomado en cuenta en este proyecto.

Finalmente, el último bloque descrito en la Figura 2, que es la conversación

con el bot, se realiza de manera conjunta con los códigos HTML, CSS y JavaScript (Apéndice C, D y E), sin embargo, en esta sección sólo se trata lo referente al código en Python, lo cuál se logra mediante la creación de un par de métodos, el primero se muestra en el Código 11:

```
@app.route("/")
def home():
    return render_template("index.html")
```

**Código 11.** Método para cargar HTML.

Este primer método es el que carga el archivo de la página web escrita en el código HTML que se detallará más adelante, y es indispensable para poder visualizar el chatbot.

El segundo método es el que se muestra en el Código 12:

```
@app.route("/get")
def get_bot_responses():
    userText = request.args.get("msg")
    return str(bot.get_response(respuesta))
```

**Código 12.** Método para obtener respuestas.

Este es el encargado de obtener los mensajes escritos en la página web y devolver una respuesta acorde a lo obtenido. Cabe recalcar que este método hace uso de la ruta “/get”, por lo que se debe utilizar esta misma ruta para guardar los mensajes escritos por el usuario, de otra manera, el bot no funcionará.

- **Código HTML**

En este código se trata la disposición de elementos que se visualizarán en el chatbot, para lograr una apariencia agradable para el usuario se vincula un documento CSS por medio de la instrucción del Código 13:

```
<link rel="stylesheet" type="text/css" href="/static/style.css">
```

**Código 13.** Enlace a archivo CSS.

De la misma manera, se vincula un documento de JavaScript para manejar el comportamiento de la página con la instrucción del Código 14:

```
<script src="./script.js"></script>
```

**Código 14.** Enlace a archivo JavaScript.

Antes de llegar a la creación de los elementos del chatbot se formó un formulario con ayuda de etiquetas (<label>), campos de texto (<input>) y botones (<button>) en el que se pregunta nombre, edad y sexo, esto es importante para más adelante tener suficientes datos para realizar minería de datos. En la creación de los elementos del chatbot sólo se hizo uso de una etiqueta, un campo de texto y un botón.

- **Código CSS**

Este código es uno de los complementos del código HTML, y como su nombre lo indica, es el encargado de establecer los estilos visuales a los elementos establecidos en el código HTML. El fragmento más importante de este código es el que permite visualizar los mensajes escritos como si fueran una conversación y se muestra en el Código 15:

```
#textInput{  
  
width: 87%;  
  
border: none;  
  
border-bottom: 3px solid #009688;  
  
font-family: monospace;  
  
font-size: 17px;}
```

**Código 15.** Hoja de estilos del chatbot.

En este código se aplica a la entrada de texto del usuario el tamaño, borde,

redondeado, color, tipo de letra y tamaño de letra respectivamente, dentro del archivo CSS también hay asignaciones totalmente iguales para las respuestas del bot, con la excepción de que se utiliza otro color diferente.

- **Código JavaScript**

Este código es el que complementa el funcionamiento del último bloque de la Figura 2 junto con el código de Python, pues es el encargado de almacenar la información ingresada en la página web para que después pueda ser recuperada por el código de Python.

El fragmento de código que permite este almacenamiento de las respuestas es el Código 16:

```
$.get("/get", {msg: rawText, nombre: $("#nombre").val(), edad: $("#edad").val(),
sexo: $("input:radio[name=sexo]:checked").val()}).done(function(data) {

    var botHtml = "<p class='botText'><span>Capturamos su
comentario como <mark>" + data + "</mark></span></p>";

    $("#chatbox").append(botHtml);

    var TyBot = "<p class='botText'><span>Si lo desea, puede
ingresar un nuevo comentario, de lo contrario escriba la palabra
<mark>adios</mark></span></p>";

    $("#chatbox").append(TyBot);

    document.getElementById("userInput").scrollIntoView({block:
"start", behavior: "smooth"});});
```

**Código 16.** Script para almacenar respuestas.

Se puede apreciar que este método almacena varios datos en la ruta “/get” tal y como se había escrito que debía ser anteriormente. Adicionalmente se muestran líneas que contienen lenguaje HTML, las cuales sirven para poder incluir los mensajes ingresados y las respuestas obtenidas en forma de conversación.

Cabe destacar que en este fragmento de código se hace uso de la biblioteca JQuery que simplifica el manejo de los datos, para incluir esta biblioteca en el código se escribió la instrucción del Código 17:

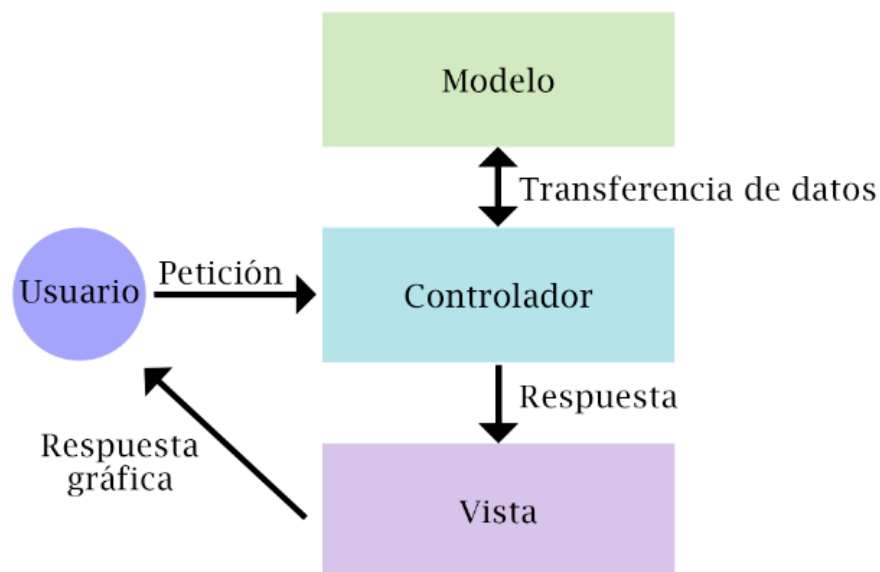
```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

**Código 17.** Enlace de JQuery.

Esta instrucción vincula la biblioteca al archivo de JavaScript, lo único que se debe hacer es indicar la dirección de la biblioteca como se muestra.

Ahora se tiene una explicación completa del desarrollo del módulo de chat, y sólo queda mencionar que todos los códigos explicados aquí (Apéndices B, C, D y E) partieron con el modelo de programación MVC en mente.

En la Figura 3 se aprecia un diagrama del modelo de programación MVC que también sirve para visualizar más fácil el resultado final de este módulo.



**Figura 3.** Diagrama sobre modelo de programación MVC.

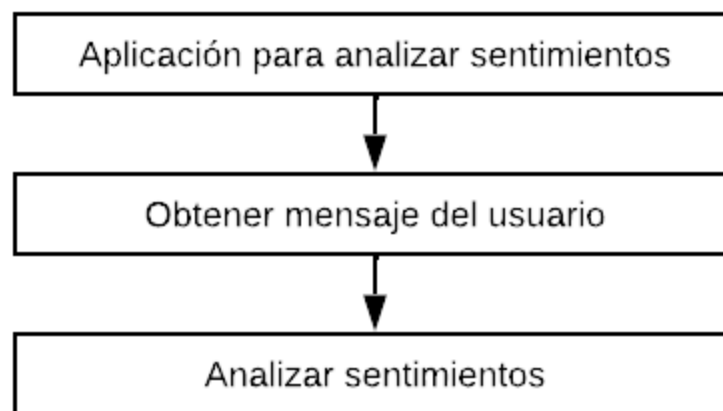
La figura 3 detalla la interacción del usuario desde un inicio con la página web:

Primero el usuario solicita acceso a la página web al controlador, que, en el caso de este proyecto, el controlador es el código de Python, mismo que garantiza el acceso y se responde

con la vista, que en este caso es el código HTML junto con el CSS, de esta manera puede responder al formulario e ingresar comentarios en el chatbot, los cuales son almacenados por el modelo que es el código de JavaScript, y es el que, de ser necesario, puede transferir datos al controlador.

## 6.2. Módulo de análisis de sentimientos

Para comprender más fácilmente el desarrollo de este módulo se realizó un diagrama de bloques, mismo que se muestra en la Figura 4.



**Figura 4.** Diagrama de bloques sobre el módulo de análisis de sentimientos.

El análisis de sentimientos de este proyecto se realizó mediante una biblioteca de Python llamada `Sentiment_Analysis_Spanish`, para su instalación se hace uso de la instrucción del Código 18 en la línea de comandos:

```
pip install sentiment-analysis-spanish
```

**Código 18.** Instalación de la biblioteca de `sentiment-analysis-spanish`.

Una vez instalada la biblioteca y para cumplir con el primer bloque de desarrollo de acuerdo con la Figura 4, se incluyó la siguiente línea mostrada en el Código 19 en el código de Python obtenido en el módulo de chat:

```
valoracion = sentiment_analysis.SentimentAnalysisSpanish()
```

**Código 19.** Inicialización del análisis de sentimientos.

Esta línea de código realiza la inicialización de un objeto con los métodos de la biblioteca que se usó para analizar sentimientos, y con lo que se cumple el primer bloque del desarrollo del módulo.

Lo siguiente es obtener el mensaje introducido por el usuario, sin embargo, esto ya se realiza por el código realizado en el módulo de chat, ya que el mensaje del usuario ya era obtenido para así poder producir una respuesta acorde de parte del chatbot, sólo es necesario guardar el mensaje en otra variable para controlar específicamente el análisis de datos, por lo que el segundo bloque del módulo se considera como completado.

Para el último bloque de desarrollo de este módulo sólo resta usar el método para realizar el análisis de sentimientos, mismo que se realiza conforme al Código 20:

```
userTextPuntaje = valoracion.sentiment(userText)
```

**Código 20.** Obtención de puntaje del análisis de sentimientos.

En esta instrucción se analiza el mensaje almacenado en la variable “userText” por medio del método sentiment, esto devuelve un valor numérico entre 0 y 1, donde un valor cercano a 0 es negativo y un valor cercano a 1 es positivo.

### **6.3. Módulo de clasificación de opiniones**

El objetivo de este módulo es el de preparar los datos generados por el módulo de análisis de sentimientos para ser ingresados en una base de datos, por lo que utiliza un criterio de clasificación de opiniones que dé sentido a las ponderaciones numéricas recopiladas para así poder distinguir entre opiniones muy negativas, negativas, neutrales, positivas y muy positivas.

La Tabla 2 muestra los rangos de las ponderaciones de los mensajes y sus significados:

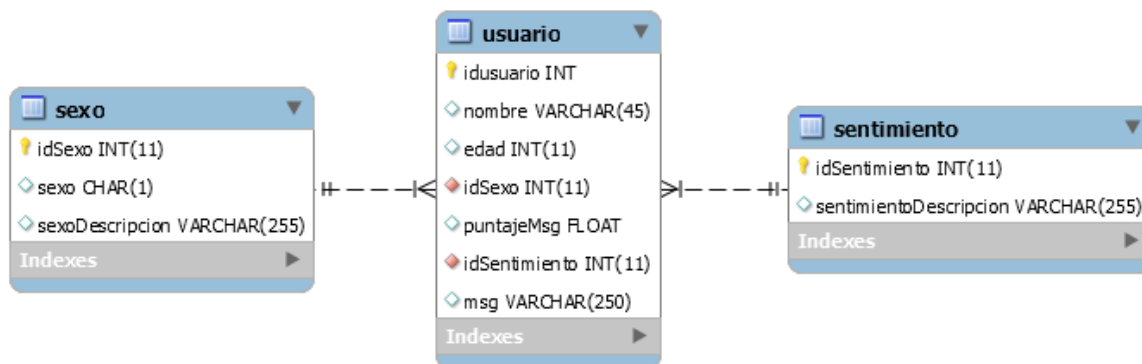
Rango de ponderación	Clasificación
0 - 0.19	Muy negativo
0.2 – 0.39	Negativo
0.4 – 0.59	Neutral
0.6 – 0.79	Positivo
0.8 - 1	Muy positivo

**Tabla 2.** Tabla de clasificación de acuerdo con el rango de ponderación numérica de un mensaje.

Para clasificar los mensajes se utilizan líneas de código en Python con distintas instrucciones if para cubrir todos los rangos previstos, de esta forma se puede codificar la respuesta en números de acuerdo con cómo se entrenó al bot en el módulo de chat, y de esta manera también se clasifica para guardar en base de datos.

#### 6.4. Base de datos

Para almacenar los datos introducidos en la página web, se creó una base de datos relacional de SQL, el modelo de la base de datos se puede observar en la Figura 5.



**Figura 5.** Modelo relacional de la base de datos.

Esta base de datos está especialmente pensada para poder realizar minería de datos sobre ella, a continuación, se detalla cada tabla perteneciente al modelo:

- **Sexo:** Esta tabla fue creada con el fin de almacenar los datos relacionados con el sexo del usuario que ingrese comentarios en el chatbot. Sus campos son los siguientes:
  - **idSexo:** Llave primaria de tipo entero con capacidad de once dígitos que almacena el valor índice del registro.
  - **sexo:** Campo con capacidad para un solo carácter con la finalidad de escribir la letra representativa del sexo.
  - **sexoDescripcion:** Campo de tipo VARCHAR con capacidad de 255 caracteres que sirve para agregar una descripción de cada sexo.

Por defecto esta tabla tiene los valores que se muestran en la Tabla 3.

idSexo	sexo	sexoDescripcion
1	M	Mujer
2	H	Hombre
99	N	Se distinguen solo dos generos para fines de registro

**Tabla 3.** Valores por defecto de la tabla sexo.

- **Sentimiento:** Esta tabla fue creada con el fin de almacenar los datos relacionados con el sentimiento detectado en el mensaje ingresado por un usuario al chatbot. Sus campos son los siguientes:
  - **idSentimiento:** Llave primaria de tipo entero con capacidad de once dígitos que almacena el valor índice del registro.
  - **sentimientoDescripcion:** Campo de tipo VARCHAR con capacidad de 255 caracteres que sirve para agregar una descripción de cada sentimiento.

Por defecto esta tabla tiene los valores que se muestran en la Tabla 4.

idSentimiento	sentimientoDescripcion
1	Muy negativo
2	Negativo
3	Neutro
4	Positivo
5	Muy positivo

**Tabla 4.** Valores por defecto de la tabla sentimiento.

- Usuario: Esta tabla fue creada con el fin de almacenar todos los datos que el usuario ha ingresado al chatbot. Sus campos son los siguientes:
  - idusuario: Llave primaria de tipo entero que almacena el valor índice del registro.
  - nombre: Campo de tipo VARCHAR con capacidad de 45 caracteres que sirve para almacenar el nombre del usuario.
  - edad: Campo de tipo entero con capacidad de once dígitos que almacena la edad del usuario.
  - idSexo: Llave foránea que sirve de vínculo entre la tabla usuario y la tabla sexo, es de tipo entero con capacidad de once dígitos que almacena el valor índice del registro de la tabla sexo.
  - puntajeMsg: Campo de tipo flotante que sirve para almacenar la ponderación numérica resultante de analizar los sentimientos del mensaje del usuario.
  - IdSentimiento: Llave foránea que sirve de vínculo entre la tabla usuario y la tabla sentimiento, es de tipo entero con capacidad de once dígitos que almacena el valor índice del registro de la tabla sentimiento.
  - msg: Campo de tipo VARCHAR con capacidad de 255 caracteres que sirve para almacenar el mensaje del usuario.

Para insertar en esta base de datos (Apéndice A) se necesitó crear una clase con un método en el código de Python (Apéndice B) y es el mostrado en el Código 21:

```
class DataBase:

    def insertar(self, name, edad, sexo, pnt, senti, msg):

        sql = "insert into usuario values (NULL, '{}', {}, {}, {}, {}, '{})".format(name, edad,
sexo, pnt, senti, msg)
```

**Código 21.** Método para insertar en BD.

Este método que se encuentra en la clase “DataBase” ejecuta una instrucción para insertar los datos recibidos en la base de datos, lo anterior sólo es la declaración del método, por lo que para hacer que se ejecute, se requiere hacer uso es la función recopilando todos los datos necesarios, la ejecución de este método se ejecuta con las líneas del Código 22:

```
database = DataBase()

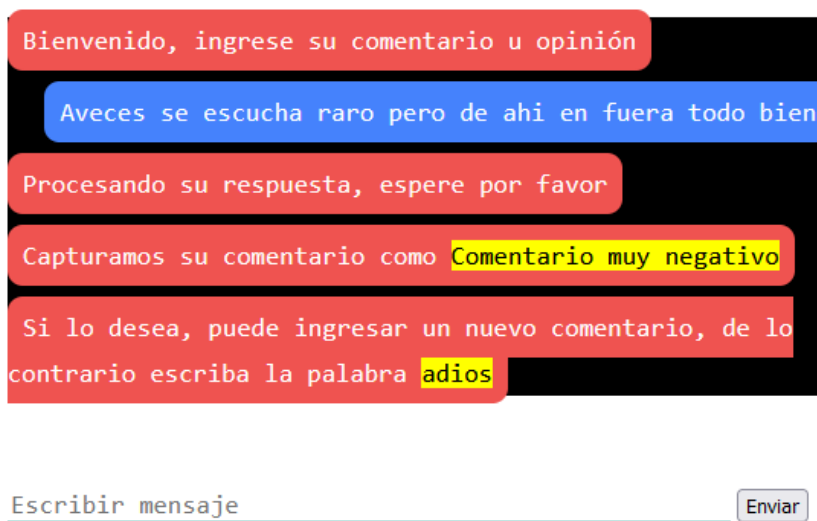
database.insertar(userTextName, userTextEdad, userTextSexo, userTextPuntaje,
userTextSentiment, userText)
```

**Código 22.** Inicialización de objeto DataBase.

## 7. Resultados

A continuación, se muestran ejemplos de la ejecución en los que se obtienen diversas etiquetas de acuerdo con la clasificación de los mensajes. Además de esto, cabe mencionar que se insertaron 200 opiniones a la base de datos, mismas que fueron extraídas de una página de venta de audífonos de Amazon [27].

La Figura 6 muestra la pantalla de ejecución en la que se obtiene una clasificación muy negativa.



**Figura 6.** Clasificación muy negativa de mensaje.

La Figura 7 muestra la pantalla de ejecución en la que se obtiene una clasificación negativa.

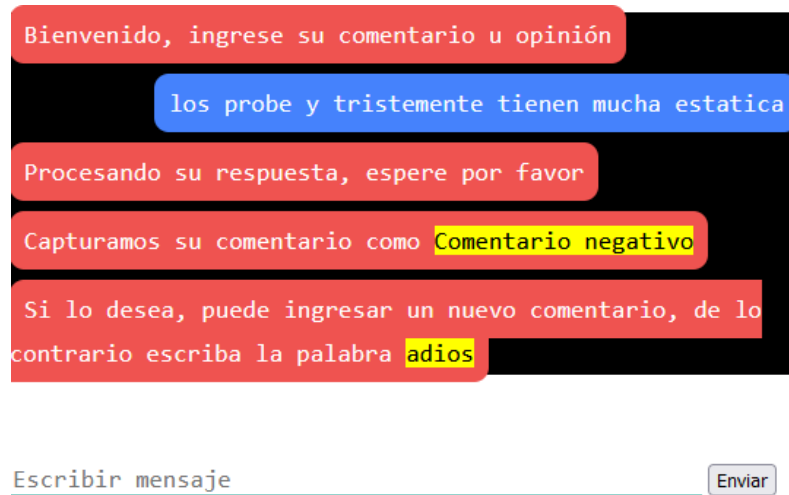


Figura 7. Clasificación negativa de mensaje.

La Figura 8 muestra la pantalla de ejecución en la que se obtiene una clasificación neutra.

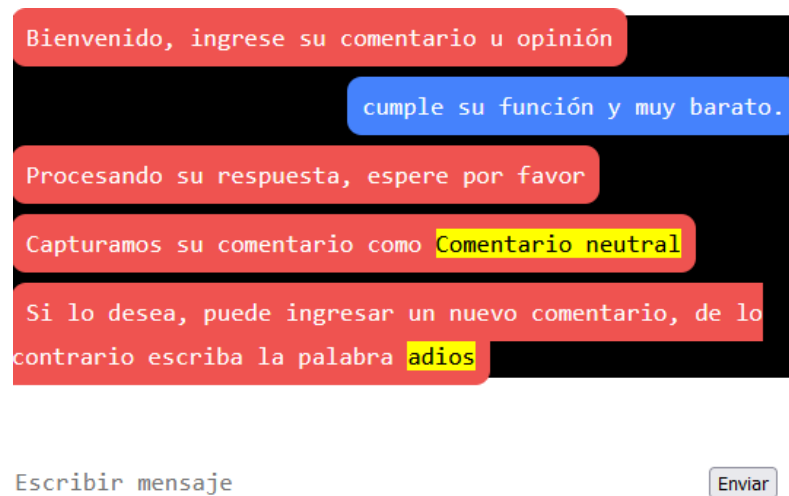


Figura 8. Clasificación neutra de mensaje.

La Figura 9 muestra la pantalla de ejecución en la que se obtiene una clasificación positiva.

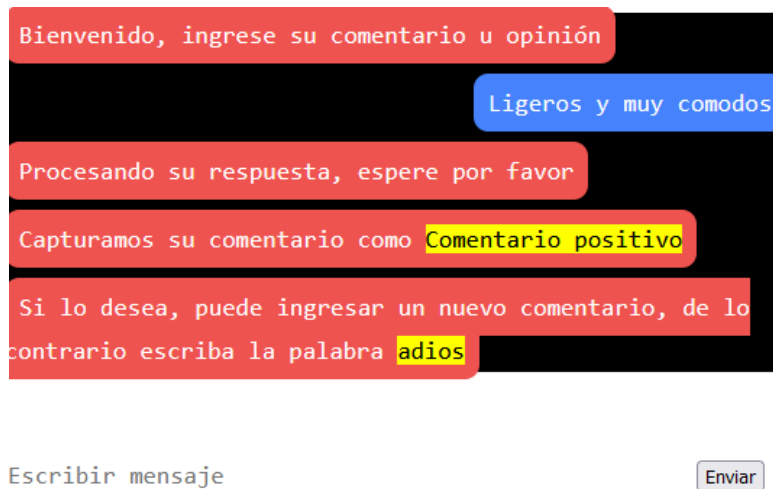


Figura 9. Clasificación positiva de mensaje.

La Figura 10 muestra la pantalla de ejecución en la que se obtiene una clasificación muy positiva.

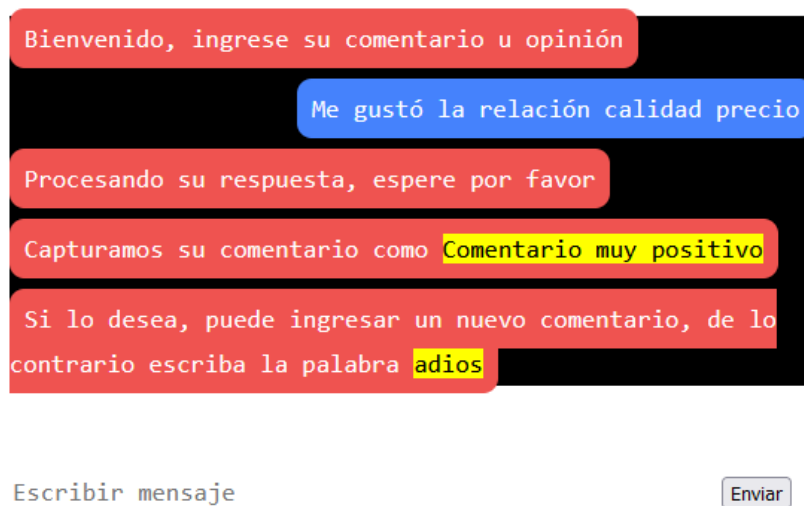


Figura 10. Clasificación muy positiva de mensaje.

## 8. Análisis y discusión de resultados

De acuerdo con los resultados obtenidos, podemos dividir los resultados en dos secciones:

### 8.1. Resultados relacionados con el análisis de sentimientos

Los resultados obtenidos relacionados con el análisis de sentimientos son en su mayoría satisfactorios, con aproximadamente el 75% de clasificaciones correctas de acuerdo a lo registrado en la base de datos, lo cual corresponde a lo esperado de acuerdo con [21], sin embargo ocurren casos en los que evidentemente la clasificación es incorrecta, tal es el caso que se puede observar con la Figura 11 y la Figura 12, donde se clasifica el mensaje “malo” como muy negativo, y el mensaje “muy malo” como negativo, el resultado esperado en este caso está invertido, esto es causa de la biblioteca empleada para el análisis de sentimientos, que si bien puede clasificar correctamente gran parte de los comentarios procesados, este resultado deja ver que la clasificación no es 100% certera, por lo que se requiere seguir entrenando a la biblioteca.

La Figura 11 muestra la pantalla de ejecución en la que se obtiene una clasificación muy negativa.

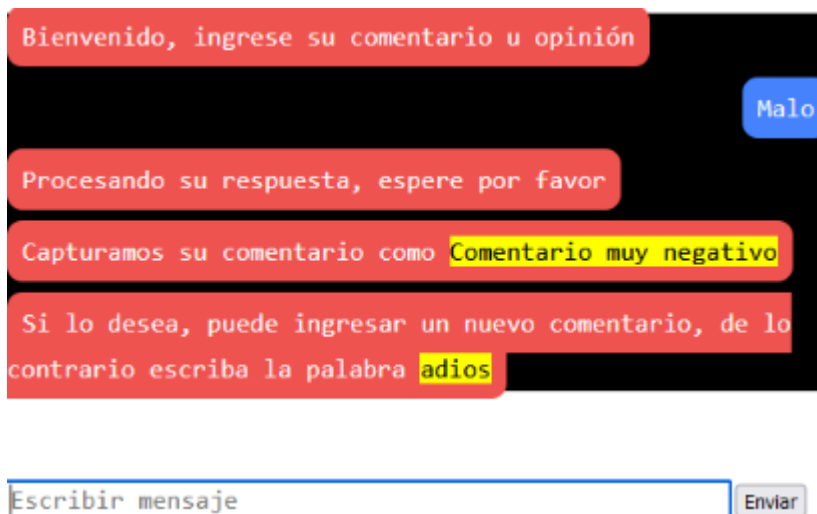


Figura 11. Comentario muy negativo.

La Figura 12 muestra la pantalla de ejecución en la que se obtiene una clasificación negativa.

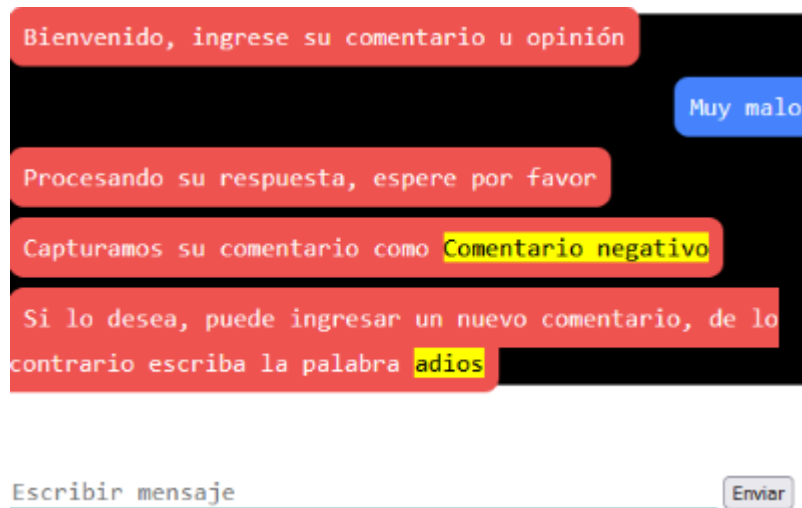
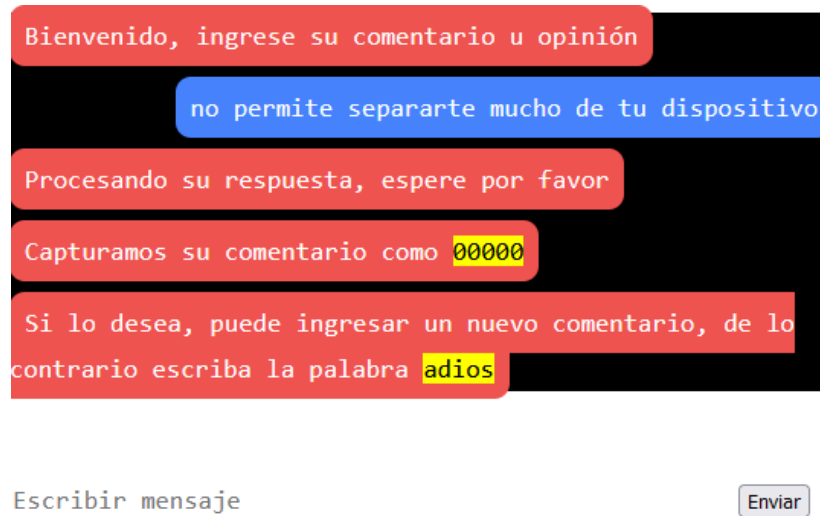


Figura 12. Comentario negativo.

Otro resultado peculiar es el que se da en la Figura 13, el cual no tiene que ver con el análisis de sentimientos, sino con un problema relacionado con la biblioteca empleada para realizar el chatbot, se puede apreciar que se arroja un resultado en forma de código, esto es debido a que por alguna razón hay una confusión en el bot respecto a las oraciones con las que fue entrenado para brindar respuestas, sin embargo, este tipo de respuesta no causa ningún error en otros procedimientos y de acuerdo con la base de datos, se ha comprobado que el dato se registra correctamente.

La Figura 13 muestra la pantalla de ejecución en la que se obtiene una clasificación errónea.



**Figura 13.** Clasificación errónea de mensaje.

## 8.2. Resultados relacionados con la minería de datos

A los datos obtenidos de las opiniones ingresadas en la base de datos adicionalmente se les realizó una minería de datos, específicamente a los campos edad, idSexo, idSentimiento, estos campos son especialmente importantes, ya que tienen una clasificación numérica a excepción del campo edad.

Se realizaron pruebas de minería de datos para tres volúmenes de datos diferentes, la interpretación de cada figura se muestra a continuación:

La Figura 14 muestra la aplicación de la técnica de clúster de minería de datos para 50 registros almacenados en la base de datos.

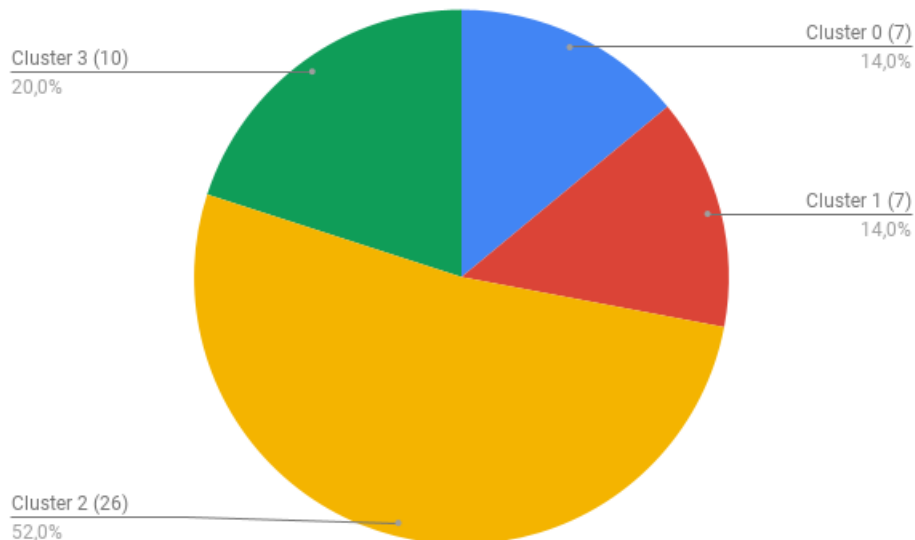
Final cluster centroids:

Attribute	Full Data (50.0)	Cluster#			
		0 (7.0)	1 (7.0)	2 (26.0)	3 (10.0)
edad	45.16	25	57	43.7308	54.7
idSexo	1.52	1	1	2	1
idSentimiento	2.96	2.2857	2.1429	2.6154	4.9

**Figura 14.** Minería de datos por clúster para 50 registros de datos.

En esta figura se puede observar en la parte de arriba el total de datos analizados (50), así como la cantidad de grupos en que se separó la información, los cuales se numeran del 0 al 3, de lado izquierdo se encuentran los campos seleccionados de la base de datos y finalmente los datos de cada columna corresponden a los resultados de agrupación de métodos estadísticos de esta técnica de minería de datos, los cuales serán explicados con uso de la Gráfica 1.

La Gráfica 1 muestra la distribución de los datos obtenidos en la Figura 14.



**Gráfica 1.** Distribución de los datos obtenidos en la Figura 14.

En la Gráfica 1 se muestran 4 clúster, mismos que pueden ser interpretados con ayuda de los datos obtenidos de la Figura 14 de la siguiente manera:

- Cluster 0: Simboliza al 14% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 25 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos neutro
- Cluster 1: Simboliza al 14% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 57 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos negativo
- Cluster 2: Simboliza al 52% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 44 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos neutro
- Cluster 3: Simboliza al 20% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 55 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos muy positivo

La Figura 15 muestra la aplicación de la técnica de clúster de minería de datos para 100 registros almacenados en la base de datos.

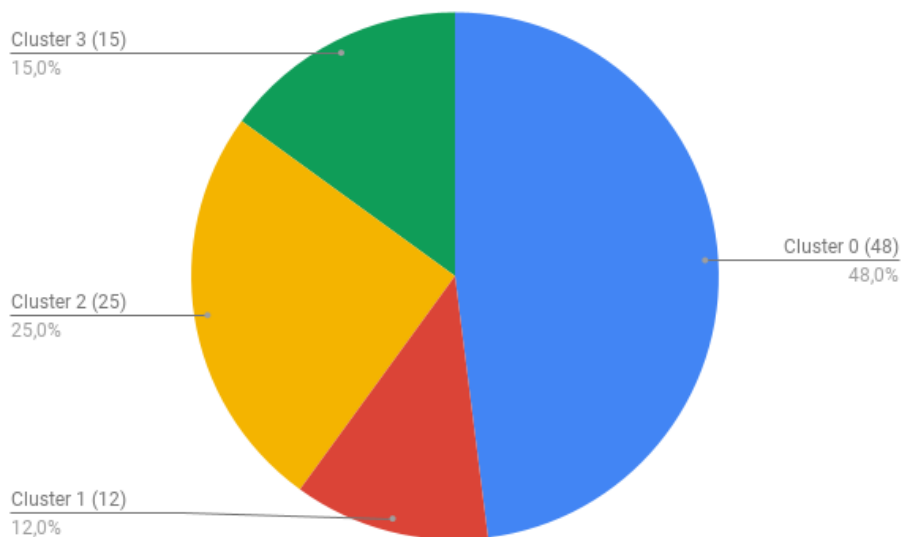
Final cluster centroids:

Attribute	Full Data (100.0)	Cluster#			
		0 (48.0)	1 (12.0)	2 (25.0)	3 (15.0)
edad	43.64	43.625	52.9167	48.32	28.4667
idSexo	1.52	1	2	2	2
idSentimiento	3.08	3.1667	1.1667	4.36	2.2

**Figura 15.** Minería de datos por clúster para 100 registros de datos.

En esta figura, tal como en la anterior, se puede observar la misma distribución de datos, sin embargo, esta vez se realiza para 100 datos analizados. Para la explicación de esta figura se hará uso de la Gráfica 2.

La Gráfica 2 muestra la distribución de los datos obtenidos en la Figura 15.



**Gráfica 2.** Distribución de los datos obtenidos en la Figura 15.

En la Gráfica 2 se muestran 4 clúster, mismos que pueden ser interpretados con ayuda de los datos obtenidos de la Figura 15 de la siguiente manera:

- Cluster 0: Simboliza al 48% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 44 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos neutro
- Cluster 1: Simboliza al 12% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 53 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos muy negativo
- Cluster 2: Simboliza al 25% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 48 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos positivo
- Cluster 3: Simboliza al 15% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 28 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos negativo

La Figura 16 muestra la aplicación de la técnica de clúster de minería de datos para 200 registros almacenados en la base de datos.

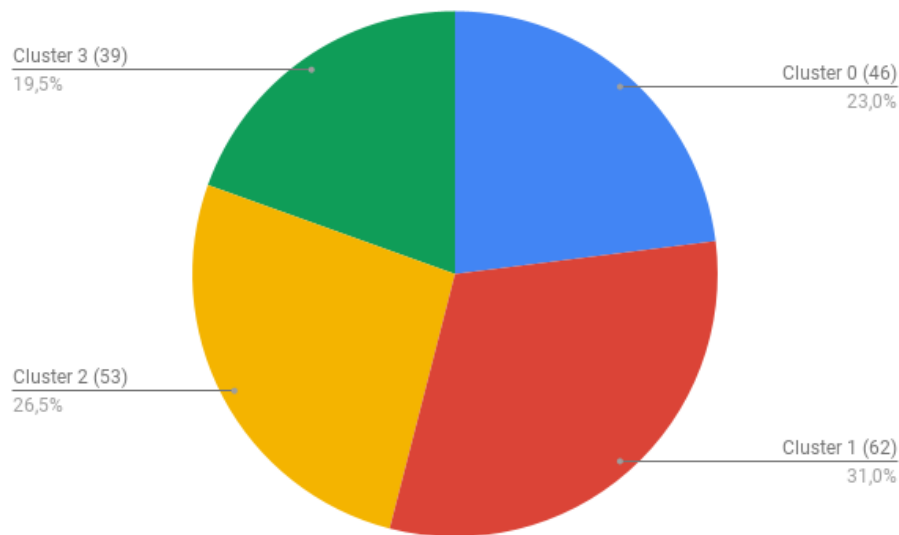
```
Final cluster centroids:
```

Attribute	Full Data (200.0)	Cluster#			
		0 (46.0)	1 (62.0)	2 (53.0)	3 (39.0)
edad	44.7	51.1957	43.2903	40.1509	45.4615
idSexo	1.505	1	2	1	2
idSentimiento	3.075	4.4565	2.0484	1.9434	4.6154

**Figura 16.** Minería de datos por clúster para 200 registros de datos.

En esta figura, tal como en la anterior, se puede observar la misma distribución de datos, sin embargo, esta vez se realiza para 200 datos analizados. Para la explicación de esta figura se hará uso de la Gráfica 3.

La Gráfica 3 muestra la distribución de los datos obtenidos en la Figura 16.



**Gráfica 3.** Distribución de los datos obtenidos en la Figura 16.

En la Gráfica 3 se muestran 4 clúster, mismos que pueden ser interpretados con ayuda de los datos obtenidos de la Figura 16 de la siguiente manera:

- Cluster 0: Simboliza al 23% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 51 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos positivo
- Cluster 1: Simboliza al 31% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 43 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos negativo

- Cluster 2: Simboliza al 26.5% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 40 años
  - Pertenecen al género femenino
  - Realizaron comentarios con un análisis de sentimientos negativo
- Cluster 3: Simboliza al 19.5% de las opiniones totales y que cuentan con las siguientes características:
  - Tienen una media de 45 años
  - Pertenecen al género masculino
  - Realizaron comentarios con un análisis de sentimientos muy positivo

## 9. Conclusiones

Este proyecto es una herramienta útil para personas u organizaciones que brindan bienes o servicios, ya que, como se pudo ver en la sección de análisis y discusión de resultados, al realizar minería de datos sobre los datos recopilados se puede obtener información muy valiosa, pudiendo obtener datos precisos sobre el grupo de personas que comentan positivamente una publicación de venta de algún artículo, permitiendo así a las empresa u organizaciones poderse enfocar en ciertas estrategias de marketing para alcanzar mejores resultados con respecto a su público objetivo.

Algo a destacar sobre este proyecto con respecto a la minería de datos, es que los resultados obtenidos se obtuvieron de volúmenes relativamente bajos de información, por lo que, de ser usado en una escala mayor, arrojaría mejores resultados.

Como trabajo a futuro se espera mejorar la certeza de clasificación del módulo de análisis de sentimientos, así como la posibilidad de poder ingresar comentarios u opiniones en otros formatos, como es el caso de opiniones por voz.

También podría considerarse agregar nuevas variables para su estudio a la base de datos, para así poder tener aún más información útil para el análisis. Esto tendría que estudiarse con cuidado para no llegar a un grado intrusivo o que haga que los usuarios se sientan incómodos con su uso.

## 10. Referencias

- [1] (s.f). Oracle.com [En línea]. Disponible: <https://www.oracle.com/mx/chatbots/what-is-a-chatbot/>
- [2] (s.f). Instituto de Ingeniería del Conocimiento [En línea]. Disponible: <https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>
- [3] (2018) IEBS [En línea]. Disponible: <https://www.iebschool.com/blog/data-mining-mineria-datos-big-data/>
- [4] L. E. Poot Terán, “Minería de opiniones sobre textos en Twitter” Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2018.
- [5] V. H. Rojas Luis, “Sistema para la clasificación de opiniones generadas en Twitter usando redes neuronales artificiales.” Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2018.
- [6] J. F. Paniagua Reyes, “Algoritmos de aprendizaje automático para el análisis de opiniones a partir de textos en español” Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2018.
- [7] S. A. Morzán Fuentes, “Detección de estados de ánimo mediante sentiment analysis en hispanohablantes”, Licenciatura, Universidad de Lima, Diciembre 2019. [En línea]. Disponible: [https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/11318/Morzán\\_Fuentes\\_Stephany.pdf?sequence=1&isAllowed=y](https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/11318/Morzán_Fuentes_Stephany.pdf?sequence=1&isAllowed=y)
- [8] R. R. López Barbosa, “Aplicación del Análisis de Sentimientos a la Evaluación de Datos Generados en Medios Sociales”, Tesis doctoral, Universidad de Alcalá, 2015. [En línea]. Disponible: <https://core.ac.uk/download/pdf/58911462.pdf>
- [9] J. León Martínez, “Identificación de depresión mediante el análisis de sentimientos”, Licenciatura, Universidad de Extremadura, 2019. [En línea]. Disponible: [https://dehesa.unex.es/bitstream/10662/9538/1/TFGUEX\\_2019\\_Leon\\_Martinez.pdf](https://dehesa.unex.es/bitstream/10662/9538/1/TFGUEX_2019_Leon_Martinez.pdf)
- [10] (2015) IEBS [En línea]. Disponible: <https://www.iebschool.com/blog/que-es-lean-manufacturing-negocios-internacionales/>
- [11] “Tipos de chatbot: ventajas y características.”, (2020). Centribal [En línea]. Disponible: <https://centribal.com/es/tipos-de-chatbot-ventajas-y-caracteristicas/>
- [12] “¿Qué es la inteligencia artificial (IA)?”, (s/f). Oracle.com [En línea]. Disponible: <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>
- [13] “¿Qué es el minado de Datos o Data Mining? Técnicas y pasos a seguir.”, (2021). Thinking for Innovation [En línea]. Disponible: <https://www.iebschool.com/blog/data-mining-mineria-datos-big-data/>
- [14] “Descubre los principales beneficios del Machine Learning.”, (s/f). Iberdrola [En línea]. Disponible: <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>
- [15] “Lenguaje Natural: Definición de la tecnología.”, (2021). Elipse [En línea]. Disponible: <https://elipse.ai/blog/definicion-de-la-tecnologia-lenguaje-natural/>
- [16] “Análisis de sentimiento, ¿qué es, cómo funciona y para qué sirve?”, (2017). Itelligent [En línea]. Disponible: <https://itelligent.es/es/analisis-de-sentimiento/>

- [17] “Biblioteca informática - LOGICA DE PROGRAMACION.”, (s/f). Google.com [En línea]. Disponible: <https://sites.google.com/site/israelcortess/documentos/biblioteca-informatica-3>
- [18] (s/f). Python.org [En línea]. Disponible: <https://www.python.org/>
- [19] “About ChatterBot — ChatterBot 1.0.8 documentation.”, (s/f). Readthedocs.io [En línea]. Disponible: <https://chatterbot.readthedocs.io/en/stable/>
- [20] “Welcome to flask — flask documentation (2.1.X).”, (s/f). Palletsprojects.com [En línea]. Disponible: <https://flask.palletsprojects.com/en/2.1.x/>
- [21] “Sentiment-spanish”, (s/f). github.com [En línea]. Disponible: <https://github.com/sentiment-analysis-spanish/sentiment-spanish>
- [22] “HTML: Lenguaje de etiquetas de hipertexto.”, (s/f). Mozilla.org [En línea]. Disponible: <https://developer.mozilla.org/es/docs/Web/HTML>
- [23] “CSS”, (s/f). Mozilla.org [En línea]. Disponible: <https://developer.mozilla.org/es/docs/Web/CSS>
- [24] “JavaScript”, (s/f). Mozilla.org [En línea]. Disponible: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [25] “JS Foundation-js. foundation.”, (s/f). JQuery [En línea]. Disponible: <https://jquery.com/>
- [26] “Weka 3 - data mining with open source machine learning software in java.”, (s/f.). Waikato.ac.nz [En línea]. Disponible: <https://www.cs.waikato.ac.nz/ml/weka/>
- [27] “Opiniones de clientes audífonos KAMYSEN.”, (s/f.). Amazon.com.mx [En línea]. Disponible: [https://www.amazon.com.mx/product-reviews/B08NJ7L2WT/ref=acr\\_dp\\_hist\\_1?ie=UTF8&filterByStar=one\\_star&reviewerType=all\\_reviews#reviews-filter-bar](https://www.amazon.com.mx/product-reviews/B08NJ7L2WT/ref=acr_dp_hist_1?ie=UTF8&filterByStar=one_star&reviewerType=all_reviews#reviews-filter-bar)

## 11. Apéndices

### A. Script SQL de la base de datos

```
drop schema proyectoTerminal;
```

```
-- BASE DE DATOS -----  
create database proyectoTerminal;  
use proyectoTerminal;  
  
-- TABLA Sexo -----  
create table Sexo (  
    idSexo int auto_increment not null,  
    sexo char,  
    sexoDescripcion varchar(255) default 'Sexo o genero ',  
    primary key(idSexo)  
);  
  
-- TABLA Sentimiento -----  
create table Sentimiento (  
    idSentimiento int auto_increment not null,  
    sentimientoDescripcion varchar(255) default 'Descripcion ',  
    primary key(idSentimiento)  
);  
  
-- TABLA Usuario -----  
create table usuario (  
    idUsuario int auto_increment not null,  
    nombre varchar(255),  
    edad int,  
    -- FK  
    idSexo int,  
  
    puntajeMsg Float,  
  
    -- FK  
    idSentimiento int,
```

```

msg varchar(255),
-- LLAVE PRIMARIA
primary key (idUserio),
-- LLAVE FORANEA
constraint FkSexo foreign key (idSexo) references Sexo(idSexo),
constraint FkSentimiento foreign key (idSentimiento) references Sentimiento(idSentimiento)
);

-- Valores de la tabla Sexo -----

insert into sexo values (NULL,'M',"Mujer"),(NULL, 'H',"Hombre"),(99, 'N',"Se distinguen solo dos
generos para fines de registro");

-- Valores de la tabla Sentimiento -----

insert into sentimiento values (NULL,'Muy
negativo'),(NULL,'Negativo'),(NULL,'Neutro'),(NULL,'Positivo'),(NULL,'Muy positivo');

-- Registro de opiniones en la tabla usuario ----
insert into usuario values (NULL, 'Juan', '23', '2', '0.10602', '1', 'Malo');
insert into usuario values (NULL, 'Mariel', '65', '1', '0.234403', '2', "Muy malo");
insert into usuario values (NULL, 'Javier', '32', '2', '0.562806', '3', 'Bueno');
insert into usuario values (NULL, 'Fernanda', '24', '1', '0.617922', '4', 'Mejor');
insert into usuario values (NULL, 'Raúl', '20', '99', '0.962244', '5', 'Muy bueno');

```

## B. Script de ChatBot

```
# Bibliotecas
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
from sentiment_analysis_spanish import sentiment_analysis
import pymysql

# Definición de la clase para conexión a base de datos
class DataBase:
    def __init__(self):
        self.connection = pymysql.connect(
            host="localhost",
            user="root",
            password="9890915",
            db="proyectoterminal"
        )

        self.cursor = self.connection.cursor()

        print("ACCESO A BASE DE DATOS EXITOSO")

    # Definición de funciones de la clase correspondiente a base de datos
    def insertar(self, name, edad, sexo, pnt, senti, msg):
        sql = "insert into usuario values (NULL, '{}', {}, {}, {}, '{}')".format(name, edad, sexo, pnt,
senti, msg)

        try:
            self.cursor.execute(sql)
            self.connection.commit()
        except Exception as e:
            raise

    def close(self):
        self.connection.close()

database = DataBase()

app = Flask(__name__)

# Configuración de almacenamiento de las conversaciones
bot = ChatBot("ChatterBot",
    storage_adapter="chatterbot.storage.SQLStorageAdapter"
)

# Listas de entrenamiento del chatbot
trainer = ListTrainer(bot)

trainer.train([
```

```
    '00000',  
    'Comentario muy negativo'  
  ]  
)
```

```
trainer.train([  
  '00001',  
  'Comentario negativo',  
])
```

```
trainer.train([  
  '00002',  
  'Comentario neutral'  
])
```

```
trainer.train([  
  '00003',  
  'Comentario positivo'  
])
```

```
trainer.train([  
  '00004',  
  'Comentario muy positivo'  
])
```

```
# Función para cargar el archivo HTML  
@app.route("/")  
def home():  
    return render_template("index.html")
```

```
# Función para obtener las respuestas del chatbot  
@app.route("/get")  
def get_bot_responses():  
    userText = request.args.get("msg")  
    userTextName = request.args.get("nombre")  
    userTextEdad = request.args.get("edad")  
    TextSexo = request.args.get("sexo")  
    userTextSexo = clasificarSexo(TextSexo)  
    valoracion = sentiment_analysis.SentimentAnalysisSpanish()  
    userTextPuntaje = valoracion.sentiment(userText)  
    respuesta = clasificar(userTextPuntaje)  
    userTextSentiment = clasificarSent(userTextPuntaje)  
    database.insertar(userTextName, userTextEdad, userTextSexo, userTextPuntaje,  
userTextSentiment, userText)  
    # database.close()  
    return str(bot.get_response(respuesta))
```

```
# Función para clasificar sexo para formato de base de datos  
def clasificarSexo(mensaje):  
    if mensaje == "Femenino":  
        return 1
```

```
elif mensaje == "Masculino":  
    return 2  
else:  
    return 99
```

# Función para clasificar sentimiento para formato de base de datos

```
def clasificarSent(mensaje):  
    # Mensaje muy negativo  
    if mensaje < 0.2:  
        return 1  
    # Mensaje negativo  
    elif 0.2 <= mensaje < 0.4:  
        return 2  
    # Mensaje neutro  
    elif 0.4 <= mensaje < 0.6:  
        return 3  
    # Mensaje positivo  
    elif 0.6 <= mensaje < 0.8:  
        return 4  
    # Mensaje muy positivo  
    elif 0.8 <= mensaje <= 1:  
        return 5
```

# Función clasificar puntaje de positividad de acuerdo al mensaje del usuario para que el bot pueda responder

```
def clasificar(mensaje):  
    # Mensaje muy negativo  
    if mensaje < 0.2:  
        return "00000"  
    # Mensaje negativo  
    elif 0.2 <= mensaje < 0.4:  
        return "00001"  
    # Mensaje neutro  
    elif 0.4 <= mensaje < 0.6:  
        return "00002"  
    # Mensaje positivo  
    elif 0.6 <= mensaje < 0.8:  
        return "00003"  
    # Mensaje muy positivo  
    elif 0.8 <= mensaje <= 1:  
        return "00004"
```

```
if __name__ == "__main__":  
    app.run()
```

## C. Página con HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <link rel="stylesheet" type="text/css" href="/static/style.css">
  <title>ChatBot UAM AZC</title>

</head>
<body>

  <h1>Chatbot</h1>
  <div>
    <div id="formReg">
      <!--Construccion de formulario -->
      <h4>Formulario de registro</h4>
      <input class="controls" type="text" name="nombre" id="nombre"
placeholder="Ingresar nombre">
      <form class="formularios">
        <label for="edad">Edad: </label>
        <input type="number" id="edad" placeholder="Ingresar edad"><br>
      </form>
      <form class="formularios">
        <label>Sexo: </label><br>
        <input type="radio" name="sexo" value="Masculino">Masculino<br>
        <input type="radio" name="sexo" value="Femenino">Femenino<br>
        <input type="radio" name="sexo" value="Sin especificar" checked>Sin
especificar<br>
      </form>
      <button class="buttons" type="button" onclick="ocultarForm();">Registrarse</button>
    </div>
    <!--Construccion de chatbox -->
    <div id="cajaDeChat">
      <div id="chatbox">
        <p class="botText"><span>Bienvenido, ingrese su comentario u
opinión</span></p>
      </div>
      <div id="userInput">
        <input id="textInput" type="text" name="msg" placeholder="Escribir mensaje">
        <input id="buttonInput" type="submit" value="Enviar">
      </div>
    </div>
  </div>
</body>
</html>
```

## D. Hoja de estilo en cascada (CSS)

```
body{
  font-family: sans-serif;
  background-color: white;
}

h1{
  color: black;
  margin-bottom: 0;
  margin-top: 0;
  text-align: center;
  font-size: 47px;
}

#chatbox{
  background-color: black;
  margin-left: auto;
  margin-right: auto;
  width: 40%;
  margin-top: 60px;
}

#cajaDeChat{
  display: none;
}

#formReg{
  width: 400px;
  background: #24303c;
  padding: 20px;
  margin: auto;
  margin-top: 50px;
  border-radius: 4px;
  font-family: "sans-serif";
  color: white;
  box-shadow: 7px 13px 37px #000;
}

h4{
  font-size: 22px;
  margin-bottom: 40px;
  text-align: center;
}

.controls{
  width: 95%;
  background: #24303c;
  padding: 10px;
  border-radius: 4px;
  margin-bottom: 16px;
  border: 1px solid #1f53c5;
```

```

font-family: "sans-serif";
font-size: 18px;
color: white;
}

.formularios{
width: 95%;
background: #24303c;
padding: 10px;
border-radius: 4px;
margin-bottom: 3px;
font-family: "sans-serif";
font-size: 18px;
color: white;
}

.buttons{
width: 100%;
background: #1f53c5;
border: none;
padding: 12px;
color: white;
margin: 16px 0;
font-size: 16px;
}

#userInput{
margin-left: auto;
margin-right: auto;
width: 40%;
margin-top: 60px;
}

#textInput{
width: 87%;
border: none;
border-bottom: 3px solid #009688;
font-family: monospace;
font-size: 17px;
}

buttonInput{
padding: 3px;
font-family: monospace;
font-size: 17px;
}

.userText{
color: white;
font-family: monospace;
font-size: 17px;
text-align: right;
line-height: 30px;
}

```

```
}  
  
.userText span{  
  background-color: #4582FC;  
  padding: 10px;  
  border-radius: 10px;  
}  
  
.botText{  
  color: white;  
  font-family: monospace;  
  font-size: 17px;  
  text-align: left;  
  line-height: 30px;  
}  
  
.botText span{  
  background-color: #EF5350;  
  padding: 10px;  
  border-radius: 10px;  
}
```

## E. JavaScript de manejo del ChatBot

```
<script>
  <!--Funcion que solicita respuesta del chatbox -->
  function getBotResponse(){
    var rawText = $("#textInput").val();
    <!--Condicional por si se desea seeguir o terminar el proceso -->
    if(rawText == "adios" || rawText == "Adios"){
      <!--Lineas para hacer que se inserte el texto en el chatbox -->
      var userHtml = "<p class=\"userText\"><span><mark>" + rawText +
"</mark></span></p>";
      $("#chatbox").append(userHtml);

      <!--Lineas para ocultar boton de enviar y caja de texto -->
      document.getElementById("textInput").style.display = "none";
      document.getElementById("buttonInput").style.display = "none";

      var ByeBot = "<p class=\"botText\"><span>Hasta luego</span></p>";
      $("#chatbox").append(ByeBot);
    }
    else{
      <!--Lineas para hacer que se inserte el texto en el chatbox -->
      var userHtml = "<p class=\"userText\"><span>" + rawText + "</span></p>";
      $("#textInput").val();
      $("#chatbox").append(userHtml);
      document.getElementById("userInput").scrollIntoView({block: "start",
behavior: "smooth"});
      var StandByBot = "<p class=\"botText\"><span>Procesando su respuesta,
espere por favor</span></p>";
      $("#chatbox").append(StandByBot);

      <!--Funcion para guardar los datos recopilados en la ruta /get -->
      $.get("/get", {msg: rawText, nombre: $("#nombre").val(), edad:
$("#edad").val(), sexo:
$("#input:radio[name=sexo]:checked").val()}).done(function(data){
      var botHtml = "<p class=\"botText\"><span>Capturamos su comentario
como " + data + "</span></p>";
      $("#chatbox").append(botHtml);
      var TyBot = "<p class=\"botText\"><span>Si lo desea, puede ingresar un
nuevo comentario, de lo contrario escriba la palabra
<mark>adios</mark></span></p>";
      $("#chatbox").append(TyBot);
      document.getElementById("userInput").scrollIntoView({block: "start",
behavior: "smooth"});
    });
  }
}
<!--Funcion para ocultar formulario -->
```

```
function ocultarForm(){
    document.getElementById("cajaDeChat").style.display = "block";
    document.getElementById("formReg").style.display = "none";
}

```